INTERNATIONAL SERIES IN OPERATIONS RESEARCH AND MANAGEMENT SCIENCE

Fundamentals of Traffic Simulation

Jaume Barceló
Editor

Springer

International Series in Operations Research & Management Science

Volume 145

For further volumes:
http://www.springer.com/series/6161

Jaume Barceló
Editor

Fundamentals of Traffic Simulation

Springer

Editor
Jaume Barceló
Department of Statistics & Operations
Research
Universitat Politècnica de Catalunya
Jordi Girona Salgado 1-3
Campus Nord, Mòdul C5
08034 Barcelona, Spain
jaume.barcelo@upc.edu

ISSN 0884-8289
ISBN 978-1-4419-6141-9 e-ISBN 978-1-4419-6142-6
DOI 10.1007/978-1-4419-6142-6
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2010930518

"When I *use a word", Humpty Dumpty said, in rather a scornful tone, "it means just what I choose it to mean – neither more nor less."*
"The question is", said Alice, "whether you can *make words mean so many different things."*

Lewis Carroll, Through the Looking Glass, (Chapter IV, Humpty Dumpty)

"¿Qué gigantes?," dijo Sancho Panza.
"Aquellos que allí ves," respondió su amo, "de los brazos largos, que los suelen tener algunos de casi dos leguas."
"Mire vuestra merced," respondió Sancho, "que aquellos que allí se aparecen, no son gigantes sino molinos de viento, y lo que en ellos parecen brazos son las aspas, que, volteadas del viento, hacen andar la piedra del molino."

Miguel de Cervantes, El Quijote, Capítulo VIII – Del buen suceso que el valeroso Don Quijote tuvo en la espantable y jamás imaginada aventura de los molinos de viento, con otros sucesos dignos de felice recordación.[1]

[1] *"What giants?" said Sancho Panza.*
"Those you see there," answered his master, "with the long arms, and some have them nearly two leagues long."
"Look, your worship," said Sancho. "What we see there are not giants but windmills, and what seem to be their arms are the vanes that turned by the wind make the millstone go."

Miguel de Cervantes: Don Quixote (1605) Chapter VIII, Of the good fortune which the valiant Don Quixote had in the Terrible and Undreamed-of Adventure of the Windmills, with Other Occurrences Worthy to be Fitly Recorded

Preface

The increasing power of computer technologies, the evolution of software engineering and the advent of the intelligent transport systems has prompted traffic simulation to become one of the most used approaches for traffic analysis in support of the design and evaluation of traffic systems. The ability of traffic simulation to emulate the time variability of traffic phenomena makes it a unique tool for capturing the complexity of traffic systems.

In recent years, traffic simulation – and namely microscopic traffic simulation – has moved from the academic to the professional world. A wide variety of traffic simulation software is currently available on the market and it is utilized by thousands of users, consultants, researchers and public agencies. Microscopic traffic simulation based on the emulation of traffic flows from the dynamics of individual vehicles is becoming one the most attractive approaches.

However, traffic simulation still lacks a unified treatment. Dozens of papers on theory and applications are published in scientific journals every year. A search of simulation-related papers and workshops through the proceedings of the last annual TRB meetings would support this assertion, as would a review of the minutes from specifically dedicated meetings such as the International Symposiums on Traffic Simulation (Yokohama, 2002; Lausanne, 2006; Brisbane, 2008) or the International Workshops on Traffic Modeling and Simulation (Tucson, 2001; Barcelona, 2003; Sedona, 2005; Graz 2008). Yet, the only comprehensive treatment of the subject to be found so far is in the user's manuals of various software products.

The purpose of this book is to fill in the gaps and to provide practitioners and researchers with a unified, comprehensive framework for the following:

- Simulation as a well-established and grounded OR technique and its specificities when applied to traffic systems
- The main approaches to traffic simulation and the principles of traffic simulation model building
- The fundamentals of traffic flow theory and its application to traffic simulation
  - Microscopic traffic modeling
  - Mesoscopic traffic modeling
  - Macroscopic traffic modeling

- The principles of dynamic traffic assignment and its application to traffic simulation
- The calibration and validation of traffic simulation models

To achieve these goals the main traffic simulator developers have been invited to contribute a chapter in which each of them describes the following:

Their approach to model building
Which are their fundamental core models – car following, lane changing, etc. – and how they have been implemented
How they deal with dynamic traffic assignment
The proposed methodology for the calibration and validation of traffic simulation models
Which extended modeling capabilities they have with user applications

Additionally, the material is complemented by a selected overview of advanced case studies and applications.

The list of contributions is not exhaustive; it would have exceeded the planned length of the book and there are a few significant missing software packages (Dynasmart and VISTA), due to reasons beyond our control, but all major players in this game are present. Microscopic approaches are represented by VISSIM, AVENUE, Paramics, Aimsun, MITSIM, SUMO, and DRACULA. Mesoscopic approaches are represented by Dynameq and Dynamit, and METANET as a conspicuous representative of macroscopic traffic modeling.

Barcelona, Spain Jaume Barceló

Contents

Contributors

Constantinos Antoniou National Technical University of Athens, 9 Heroon Politechniou St., GR-15780 Athens, Greece, antoniou@central.ntua.gr

Ramachandran Balakrishna Caliper Corporation, 1172 Beacon Street, Suite 300, Newton, MA 02461, USA, rama@caliper.com

Jaume Barceló Department of Statistics and Operations Research and CENIT (Center for Innovation in Transport), Universitat Politècnica de Catalunya, Barcelona, Spain, jaume.barcelo@upc.edu

Moshe Ben-Akiva Massachusetts Institute of Technology, 77 Massachusetts Ave., Rm. 1-181, Cambridge, MA 02139, USA, mba@mit.edu

Jordi Casas TSS – Transport Simulation Systems, S.L, Passeig de Gràcia 12, 08007 Barcelona, Spain and Universitat de Vic, Carrer Sagrada Familia 7, 08500 Vic, Spain, casas@aimsun.com

Charisma F. Choudhury Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh, cfc@ce.buet.ac.bd

Martin Fellendorf University of Technology Graz, Rechbauerstrasse 12, 8010 Graz, Austria, martin.fellendorf@tugraz.at

Jaime L. Ferrer TSS – Transport Simulation Systems, S.L, Passeig de Gràcia 12, 08007 Barcelona, Spain, jlferrer@aimsun.com

Michael Florian University of Montreal, C.P. 6128, Succursale Centre-ville, Montreal, QC, H3C 3J7, Canada and INRO Consultants, Inc., 5160 Decarie Blvd. Suite 610, Montreal, QC, H3X 2H9, Canada, mike@inro.ca

David Garcia TSS – Transport Simulation Systems, S.L, Passeig de Gràcia 12, 08007 Barcelona, Spain, david@aimsun.com

Hisatomo Hanabusa i-Transport Lab. Co., Ltd., 1-4 Kanda-Jinbo-cho Chiyoda-ku Tokyo, 101-0051, Japan, hanabusa@i-transportlab.jp

Ryota Horiguchi i-Transport Lab. Co., Ltd., 1-4 Kanda-Jinbo-cho Chiyoda-ku Tokyo, 101-0051, Japan, rhoriguchi@i-transportlab.jp

Haris N. Koutsopoulos The Royal Institute of Technology, KTH Teknikringen 72, SE – 100 44 Stockholm, Sweden, hnk@infra.kth.se

Daniel Krajzewicz Institut für Verkehrssystemtechnik, Deutsches Zentrum für Luft- und Raumfahrt e.V., Rutherfordstr. 2, 12489 Berlin, Germany, daniel.krajzewicz@dlr.de

Masao Kuwahara Department of Human-Social Information Sciences, Graduate School of Information Sciences, Tohoku University, Aoba 6-6-06, Aramaki, Aoba-Ku, Sendai, 980-8579, Japan, kuwahara@plan.civil.tohoku.ac.jp

Ronghui Liu Institute for Transport Studies, University of Leeds, Leeds, UK, r.liu@leeds.ac.uk

Michael Mahut INRO Consultants, Inc., 5160 Decarie Blvd. Suite 610, Montreal, QC, H3X 2H9, Canada, michaelm@inro.ca

Albert Messmer Independent Engineer Groebenseeweg 2, D-82402 Seeshaupt, Germany, albert.messmer@t-online.de

Markos Papageorgiou Dynamic Systems and Simulation Laboratory, Technical University of Crete, 73100 Chania, Greece, markos@dssl.tuc.gr

Ioannis Papamichail Dynamic Systems and Simulation Laboratory, Technical University of Crete, 73100 Chania, Greece, ipapa@dssl.tuc.gr

Josep Perarnau TSS – Transport Simulation Systems, S.L, Passeig de Gràcia 12, 08007 Barcelona, Spain, josep@aimsun.com

Pete Sykes SIAS Ltd, 37 Manor Place, Edinburgh, Scotland, EH3 7EB, pete.sykes@sias.com

Tomer Toledo Technion – Israel Institute of Technology, Haifa 3200, Israel, toledo@technion.ac.il

Alex Torday TSS – Transport Simulation Systems, S.L, Passeig de Gràcia 12, 08007 Barcelona, Spain, torday@aimsun.com

Peter Vortisch Karlsruhe Institute of Technology, Institute for Transport Studies (IfV), Karlsruhe 76131, Germany, peter.vortisch@kit.edu

Yibing Wang Department of Civil Engineering, Monash University, Victoria 3800, Australia, yibing.wang@monash.edu

Qi Yang Caliper Corporation, 1172 Beacon Street, Suite 300, Newton, MA 02461, USA, qiyang@caliper.com

About the Authors

Constantinos Antoniou received the Diploma in Civil Engineering (1995) from the National Technical University of Athens (NTUA), Greece, an MS in transportation (1997) and a Ph.D. in transportation systems (2004) from the Massachusetts Institute of Technology (MIT). He is currently an assistant professor at NTUA. He has 14 years of experience in demand modeling, traffic simulation, intelligent transportation systems (ITS), and road safety projects.

Ramachandran Balakrishna is a research transportation engineer at Caliper Corporation, where he develops advanced mode choice models and dynamic traffic analysis techniques. Dr. Balakrishna earned his doctoral and master's degrees from the Massachusetts Institute of Technology and has a bachelor's degree in civil engineering from the Indian Institute of Technology, Madras, India. He has over a decade's experience with traffic simulation modeling, and he has published several times in transportation journals and conference proceedings.

Jaume Barceló is full professor at the Department of Statistics and Operations Research at the Technical University of Catalonia (UPC), holds a Ph.D. on physical sciences from the Universidad Autónoma de Barcelona and specializes in the application of optimization and simulation techniques for transportation problems. In 1985, he set up the research group that developed Aimsun, was co-founder and scientific director of TSS – Transport Simulation Systems until 2008. Since then he has been the scientific director of the Area of Information and Communication Technologies and Mobility at CENIT (Center for Innovation in Transport, www.cenit.es) at UPC.

Moshe Ben-Akiva is the Edmund K. Turner professor of civil and environmental engineering and the director of the Intelligent Transportation Program at the Massachusetts Institute of Technology (MIT). He holds a Ph.D. in transportation systems from MIT and has received honorary degrees from the University of the Aegean, the Université Lumière Lyon, and Sweden's Royal Institute of Technology. His awards include the Operations Research Society of America Transportation Science Dissertation Prize, MIT Department of Civil and Environmental Engineering Effective Teaching Award, MIT Samuel M. Seegal Prize for inspiring students, Lifetime Achievement Award of the International Association

for Travel Behavior Research, and the Jules Dupuit prize from the World Conference on Transport Research Society. He co-authored *Discrete Choice Analysis*, MIT Press; and co-edited *Recent Developments in Transport Modeling: Lessons for the Freight Sector*, Emerald Group Publishing.

Jordi Casas holds a master's degree in computer science and a Ph.D. in traffic simulation, both from the Universitat Politecnica de Catalunya (UPC) in Barcelona. Jordi started working on the Aimsun microsimulator at the UPC in 1993 and has since worked on the dynamic traffic assignment and mesoscopic simulator in Aimsun. Jordi is a founding partner and R&D director at TSS – Transport Simulation Systems. He is also a professor of statistics and operations research at the University of Vic, Spain.

Charisma Choudhury is an assistant professor in the Department of Civil Engineering at Bangladesh University of Engineering and Technology. Before joining BUET, she worked at the Massachusetts Institute of Technology, first as a graduate research assistant and then as a postdoctoral associate. She was actively involved in developing the driving behavior models of the next generation simulation (NGSIM) program of FHWA, USA.

Martin Fellendorf is a full professor at the Graz University of Technology, heading the Institute of Transport Planning and Highway Engineering. He focuses on transport modeling, traffic engineering, and traffic management. Martin Fellendorf has a degree in industrial engineering and received his Ph.D. in traffic signal optimization from the University of Karlsruhe in 1991. Together with a development team, he introduced the traffic flow simulator VISSIM in 1994. He was responsible for the traffic management group at PTV in Karlsruhe, Germany, before moving to Austria in 2005.

Jaime L. Ferrer holds a master's degree in computer science from the UPC, as well as a postgraduate degree in business administration from the Universitat Pompeu Fabra in Barcelona. From 1992 to 2001 he was assistant professor in the Department of Statistics and Operations Research at UPC. Jaime started the development of the Aimsun microsimulator at UPC in 1986 and maintained his hands-on involvement in its evolution until 2001. Jaime is a founding partner and managing director at TSS – Transport Simulation Systems.

Michael Florian earned his Ph.D. in engineering and science from Columbia University and is professor emeritus of both the Department of Computer Science and Operations Research and CIRRELT at the University of Montreal. He has published over 150 papers in refereed journals. He was awarded several distinctions, including the Robert Herman Lifetime Achievement Award, the Fellow of The Royal Society of Canada and the INFORMS Fellow Emeritus. He is a member of the TRB Network Modeling committee and is the founder and president of INRO.

David Garcia holds a master's degree in computer science from UPC. During the period 1992–1997 David was a researcher at UPC working on the user interface and parallelization of the Aimsun microsimulator. He joined TSS – Transport Simulation

Systems in 1997 and, as architect and project manager, he has overseen development of all commercial versions of Aimsun to date. David is a founding partner and technical director at TSS.

Hisatomo Hanabusa is a traffic engineer at i-Transport Lab. He earned his ME from the Chiba Institute of Technology. His specialty is traffic engineering, with particular emphasis on traffic simulation, ITS and probe data processing.

Ryota Horiguchi got his doctoral degree from the University of Tokyo in 1996. His major interests are traffic simulation, probe data processing, travel information service, and other related areas. In 2000 he launched i-Transport Lab., a venture company in the traffic and transport engineering field, and took office as CEO.

Haris N. Koutsopoulos is professor and head of the Traffic and Logistics Division at the School of Architecture and the Built Environment at KTH. He received his Sc.Eng. degree from the National Technical University of Athens, Greece, and MS and Ph.D. degrees in transportation systems from MIT. His research focuses on ITS, simulation-based dynamic traffic assignment, traffic simulation, simulation of urban rail operations, modeling driver behavior, and calibration of simulation models.

Daniel Krajzewicz born in Bydgoszcz (Poland), completed his study of computer science at the Technical University in Berlin. Since 2001, he is working for the Institute of Transportation Systems at the German Aerospace Center on traffic flow simulations, focusing on microscopic and submicroscopic models. He is one of the main developers of the "Simulation of Urban Mobility" (SUMO) traffic simulation, which was used for research on different traffic management questions, ranging from traffic surveillance, over traffic guidance to traffic forecast.

Masao Kuwahara is a professor at the Institute of Industrial Science, University of Tokyo. After receiving his Ph.D. from the University of California, Berkeley, he has accumulated 25 years of experience in research and education in traffic engineering. His major interests are dynamic network analysis, traffic simulation, highway capacity, traffic signals control, ITS, and other related areas. He has been appointed a member of various committees for ministries, local governments, and public corporations on transportation planning and traffic management. He has also served as an International Advisory Committee member of ISTTT, as well as an editor for several international journals, including *ITS Journal*, *Transportmetrica*, and *Sustainable Transportation*.

Ronghui Liu has 15 years of experience in transport research with research interests pertaining to traffic network microsimulation modeling. She is in a unique position to continue her research into the development of the DRACULA microsimulation model. Her research interests additionally cover the application of optimization techniques in the design of optimal transport policies, statistical estimation of trip patterns, modeling of information systems, multi-agent models of driver behavior, and modeling public transport operations. More recently, she has turned her attention to network reliability and methodology in predicting travel time variability.

Michael Mahut obtained his Ph.D. in operations research from the University of Montreal. Since joining INRO he has been the lead researcher behind the Dynameq DTA software package. He has co-authored several peer-reviewed articles about traffic modeling and DTA and continues to review articles for several scientific journals in the transportation field. He is a member of the Network Modeling and Traffic Flow Theory committees of the (US) Transportation Research Board.

Albert Messmer is an independent consulting engineer. His research interests include modeling, simulation, control of motorway networks and urban traffic systems, and the development of generic tools.

Markos Papageorgiou is a professor and director of the Dynamics System and Simulation Laboratory at the Technical University of Crete. His research interests include automatic control, optimization, and their application to traffic and transportation systems and water networks. He is the editor-in-chief of *Transportation Research – Part C*, a fellow of the IEEE, and the first recipient (2008) of the IEEE Outstanding ITS Research Award.

Ioannis Papamichail is an assistant professor with the Department of Production Engineering and Management, Technical University of Crete, Greece. He is the author and coauthor of several technical papers in scientific journals and conferences. His research interests include optimal control, local and global optimization techniques, and their application to traffic and transportation systems.

Josep Perarnau holds master's degrees in computer science and statistics and operations research, both from UPC in Barcelona. Josep was a researcher at UPC where he worked with Professor Jaume Barceló on different European Projects. He joined TSS – Transport Simulation Systems in 2005 and has since worked on the development of the Aimsun mesoscopic simulator.

Pete Sykes was the first programmer on the S-Paramics project with SIAS in 1986. After a spell working in A.I., networked virtual reality systems and computer-based training, he returned to SIAS in 1998 and now leads the Paramics Microsimulation Division. He is a graduate of Cambridge University with a degree in natural sciences, specializing in experimental physics.

Tomer Toledo is a senior lecturer at the Transportation Research Institute and Faculty of Civil and Environmental Engineering at the Technion – Israel Institute of Technology in Haifa, Israel. He received a Ph.D. in transportation systems from the Massachusetts Institute of Technology in 2002. Dr. Toledo's research is in the areas of large-scale traffic simulation, driver behavior and safety, and intelligent transportation systems.

Alexandre Torday holds a master's degree in civil engineering and a Ph.D. in ITS applications, both from the Swiss Federal Institute of Technology (EPFL) in Lausanne. After spending some years in academia, he joined TSS – Transport Simulation Systems in 2005 and has since worked on several simulation-based

projects with a focus on pushing the boundaries of established modeling methodologies. Alexandre is a partner and consulting director at TSS. He is also a member of the Swiss national expert group on intelligent transport systems and the International Standards Organization (ISO).

Peter Vortisch started working in the field of microsimulation as a computer science student in 1985. He developed the first PC-based version of VISSIM's academic ancestor simulation software at the University of Karlsruhe, Germany, and later implemented VISSIM 1.0, the first commercial release, in 1991. Vortisch holds a degree in computer science and a Ph.D. in civil engineering, both from the University of Karlsruhe. Today he is with PTV, the German software and consulting company, serving as the vice president responsible for product management and development of traffic engineering software tools.

Yibing Wang is a senior lecturer with the Department of Civil Engineering, Monash University, Australia. He holds a Ph.D. in automatic control from Tsinghua University. His research interests include traffic flow modeling, freeway traffic surveillance and control, urban traffic signal control, and VANETs. Dr. Wang is an associate editor for *IEEE Transactions on Intelligent Transportation Systems* and the book reviews editor of *Transportation Research Part C: Emerging Technologies*. He is a member of the International Federation of Automatic Control (IFAC) Technical Committee on Transportation Systems.

Qi Yang is the director of Traffic Systems at Caliper Corporation. He received his doctoral degree from MIT. In recent years, he has spent most of his time programming TransModeler and TransCAD – Caliper's travel demand forecasting software.

Chapter 1
Models, Traffic Models, Simulation, and Traffic Simulation

Jaume Barceló

1.1 The Concept of Model: A Scientific Approach to Systems Analysis

A methodological approach which has proved to be successful for the study of complex phenomena is the systems approach that considers the system as a whole, consisting of interconnected, complex, and functionally related components, which can be studied scientifically by using a formal representation or model of the system. At the simplest level (Mitchell, 1993), a model is a representation of something. According to the Webster's Dictionary, a system is

a. a complex unity formed of many often diverse parts subject to a common plan or serving a common purpose.
b. an aggregation or assemblage of objects joined in regular interaction or interdependence.

These acceptations of the word system reveal that it is something more than a mere addition of its parts or components. What makes the difference between a simple set of integrating components and a system is *the interaction or interdependence of the parts* and that they are serving a *common purpose*. We can therefore synthesize the concept of system as "a collection of *entities* that *act* and *interact* together *toward* the accomplishment of some logical end." It should be underscored that this concept of system assumes a *holistic* approach; in other words, it emphasizes the functional relationships between parts and wholes, it supposes that wholes cannot be reduced to the sum of their parts or, alternatively, that a system is more than the mere sum of its parts. In what follows we will use the word as it describes a useful representation of those aspects of the world concerning what we will call a traffic system.

J. Barceló (✉)
Department of Statistics and Operations Research and Center for Innovation in Transport, Universitat Politècnica de Catalunya, Barcelona, Spain
e-mail: jaume.barcelo@upc.edu

J. Barceló (ed.), *Fundamentals of Traffic Simulation*,
International Series in Operations Research & Management Science 145,
DOI 10.1007/978-1-4419-6142-6_1,

The exercise of building such representation usually pursues the objective of having a device for describing and understanding how a system works, behaves, and evolves over time. This device is used for predicting the output from a real system, under various conditions that are specified by the input data, without actually using the real system to make this prediction. Such a device is a formal representation that we call a model of the system.

To build a model of a system, we need first to acquire knowledge of the system that we could translate in terms of assumptions about how it works, assumptions that usually take the form of mathematical and/or logical relationships. These mathematical and logical relationships constitute the formal representation that we call a model of the system. To move from these assumptions and the knowledge supporting them to their formal representation in terms of a model, we need to set up a methodological framework for building models of systems.

After establishing the basic concepts on system modeling, we will discuss how we can move from the general concepts to the particular case of modeling traffic systems. A key issue at this point is to realize that there exists no such thing as "the model of a system." In other words, there is not a unique model of a system. A model is not independent of the objectives of the system's study. Quoting again from Mitchell's book, "model building implies making statements of some or all the beliefs about the real world that the model builder thinks are relevant to the problem at hand. Using the model is viewed as the logical manipulation of these beliefs to generate consequential beliefs equally or more relevant to the problem." The notion of a model, as the synthesis of beliefs about a system, considers the model-building process as a means of structuring the understanding of the reality by the model builder, and can be understood as a learning process aimed at answering questions about the modeled system, and finding solutions to the problem raised by the system.

That means that the model of a system is not independent of the problem that the model builder tries to solve, nor of the beliefs and understanding that he has of the modeled system. In other words, models are formal representations of systems designed to assist systems analysts in answering questions about the systems. Therefore depending on the problem, the analyst's beliefs, and the type of questions that need to be answered, the same system can be modeled in different ways. According to Minsky, "an object M is a model of a system S if it can provide valid answers to questions of an observer O on the system S." Traffic systems will become a relevant example of this plural nature of the modeling exercise. We will illustrate this point showing how different models of the same traffic system can be built depending on the approach taken.

As we pointed out, to build a model of a system, we must make assumptions about how it works. These assumptions will always be as good as the knowledge we have of the system, therefore we should accept that a model will only be a partial representation of reality, which means that it will contain various approximations, some of little consequence and others greater. The type of approximation made will reflect the training, experience, and personality of the analyst; the resources available – particularly in terms of time and funds; and the purpose of the study. *There is always a degree of arbitrariness present in the model-building process*.

We will be interested in formal models, that is, models which express in formal terms the relationships between the system components as identified by the systems analysis, according to the modeling hypotheses which are translated into the modeler's understanding of how the system behaves. Among the formal models, the ones interesting to us are the mathematical models, that is, models using the formalism of mathematics to express the system relationships in quantitative terms. The usual way to establish such formalism consists of establishing a relationship between attributes that characterize the entities composing the system and mathematical variables formally representing them. Variables can be classified as *decision variables*, representing the controllable aspects of the problem or courses of action, and *uncontrolled variables*, sets of parameters, coefficients, and constants which are system inputs determining the feasible alternatives for the courses of action. Those aspects that measure how well the objectives of the decision maker are achieved are called the *performance measures*, *measures of effectiveness*, or *utility functions*. When it can be formally expressed in terms of the decision variables, then it is usually called the *objective function*.

A mathematical model is then a formal representation of the system in mathematical terms. According to Ackoff (1962), the general form of a mathematical model is

$$\begin{aligned} &\text{OPT} \quad U(X, Y) \\ &\text{s.t.} \quad X \in W(Y) \end{aligned}$$

where

$$\text{Entity variables} \Leftrightarrow \text{Variables} \begin{cases} X = \{X_1, X_2, \ldots, X_n\} \text{ is the set of decision variables} \\ Y = \{Y_1, Y_2, \ldots, Y_m\} \text{ is the set of uncontrollable variables} \end{cases}$$

The relationship $X \in W(Y)$ defines the range of feasible values for the decision variables in terms of the uncontrolled variables. Usually the definition domain $W(Y)$ can be expressed in terms of mathematical expressions that limit the ranges of values that a decision variable can assume. Such mathematical expressions, called constraints, have the following general form:

$$W(Y) \equiv \{X : R_k(X, Y) \leq 0, k = 1, 2, \ldots, K\}$$

Usually the model objectives are the following:

- Find the values of the decision variables X that satisfy the constraints and optimize the utility function.
- Try to gain some understanding of how the modeled system behaves.

One of the main reasons for building mathematical models is that they are easy to manipulate. This allows for a quick exploration of the effects of changes in the inputs for the objective function. In contrast to real-life experiments, a new or

updated answer can often be obtained with a reasonable computing effort. It is these attributes which make mathematical models the workhorse of systems analysis.

1.2 The Model-Building Process: Methodological Framework

The model-building process usually begins by systems analysis, a process of knowledge acquisition that can also be interpreted in terms of an abstraction of reality, consisting of the conceptualization of the situation. According to Daellenbach (1995), this step has three major components:

1. *Elements of structure*: Aspects or components of the situation that are stable or change only very slowly in the time frame implied in the situation: physical structures, buildings, equipment, but also functional structural aspects, like properties of physical or logical components, etc.
2. *Elements of process*: Aspects of the situation that undergo change or are in a state of flux, such as ongoing activities within the structure, flow and processing of material or information, and ongoing decision making.
3. *Relationships between structure and process and between processes*: How does the structure affect or condition the processes? What things or aspects are direct or indirect results of such relationships?

A system description identifies and characterizes all relevant components, or entities of the system, including the structural and process relationships among them and how they determine or are determined by the system objectives. This description will constitute the basis of a formal representation or system model, which will be studied and used for manipulating problems that are of interest to the analyst. A system description consists in specifying the following:

1. The transformation process(es) or activities of the system
2. The boundaries of the system
3. The components and subsystems and the stable relationships between them or the structure
4. The inputs into the system from the environment
5. The outputs of the system

It is the duty of systems analysis to provide the elements for system description, that is, to identify the system components or entities, and characterize them in terms of their attributes; identify the interactions and relationships between entities; and specify the system objectives. The final goal of systems analysis is to acquire enough knowledge on the system to be able to

- formulate hypotheses on how the system works (modeling hypotheses) and
- characterize the entities' interactions and relationships (in terms of attributes whenever possible).

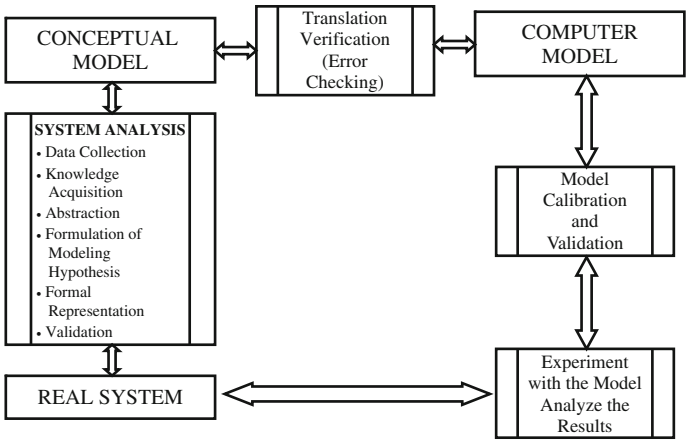


Fig. 1.1 Methodological steps of the model-building process

A methodological framework for the model-building process can be conceptualized in terms of the logic diagram depicted in Fig. 1.1. The systems analysis allows a primary representation of the system or the *conceptual model*. This conceptual model is the representation that the analyst has in his head. This is not perfect and needs validation in order to check that all main components of the system are being taken into consideration and are suitably represented in terms of their attributes. That means a refinement in the meaning of validation. Validation is then an activity that should be realized at each step of the process, and not only at the end.

In terms of a mathematical representation for which a numerical algorithm is available, translating the conceptual model can also be understood in terms of building a suitable *computer model*. With such a model, the modeling process for large systems is feasible only if suitable computing tools are available. Computer models should themselves be the object of verification, being checked regularly for errors, and validation, that is, checking that the computer model does what is expected of it to do.

The error-free computer model can then be implemented and executed to provide the solutions that will be the object of the last verification. This last verification exercise very often consists of a comparison with observed reality. The validated computer model will then become the "laboratory" for conducting suitably designed simulation experiments that will answer questions, very often "what if questions," about system behavior under the various design alternatives that configure the experimental scenarios.

How does this generic methodology apply to traffic and transportation systems? The starting point in understanding the system should be to understand what the cause of mobility is, what generates the need for mobility, and how it is satisfied. Mobility must be understood as a social and economic phenomenon, a consequence of human activities distributed across space and time. These activities generate the

need to move persons and freight between various points, which in turn generate the trips to accomplish them. The transportation system provides the infrastructures and means, ensuring that both persons and freight will be at the right location at the right time to perform the activities that will result in products and services when they are required by the market. Figure 1.2 summarizes a conceptual approach to identifying the main components of a transport system and their interrelationships. Roughly speaking, we can interpret the dynamics of the process in terms of the interactions between two major components: the system and the users. From the user's side, the main objective in building a model is to understand how travel decisions are made. This assumes that users – that is, travelers – have requirements and preferences and an assessment of how the transportation systems work, based, for example, on experience gained from the daily use of the system.

This experience supports a perception on how the system performs, which – combined with the user's objectives – is the basis of the user's decision-making process. This in turn determines his or her choices: the starting time t of the journey, the route r from the origin to the destination, and the transportation mode m to make the trip (i.e., passenger car, public transport bus, tramway, metro, and railway). Taking into account that origins and destinations are determined by the spatial distribution of socioeconomic activities (a consequence of the land use policies), the combination of all these ingredients originates in the demand for transportation, which is the first main component in the model of any transportation system.

Transportation demand can be modeled in various ways. The more detailed way consists of describing demand in terms of the activities generating it, which is accomplished by following the process that we have described, this is, the activity-based demand representation. However, so far, the most used approach to

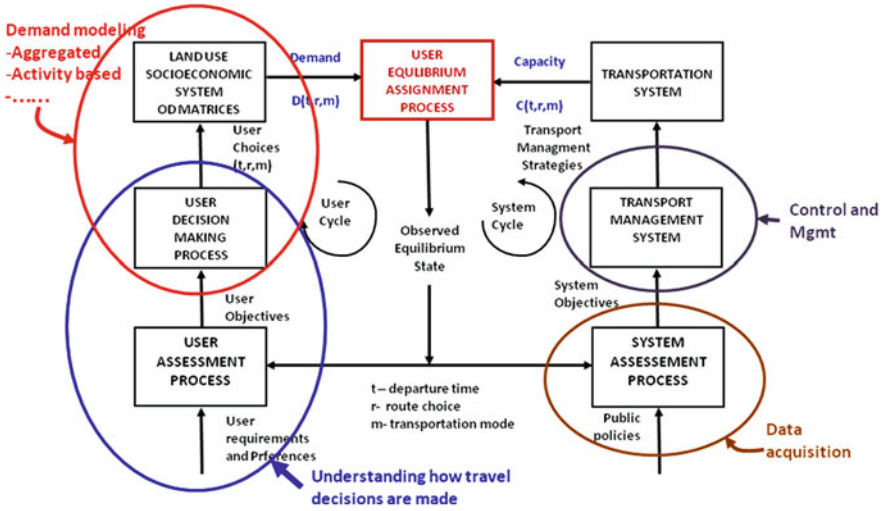


Fig. 1.2 Components of the transportation system and their interrelationships (adapted from OECD, 1987)

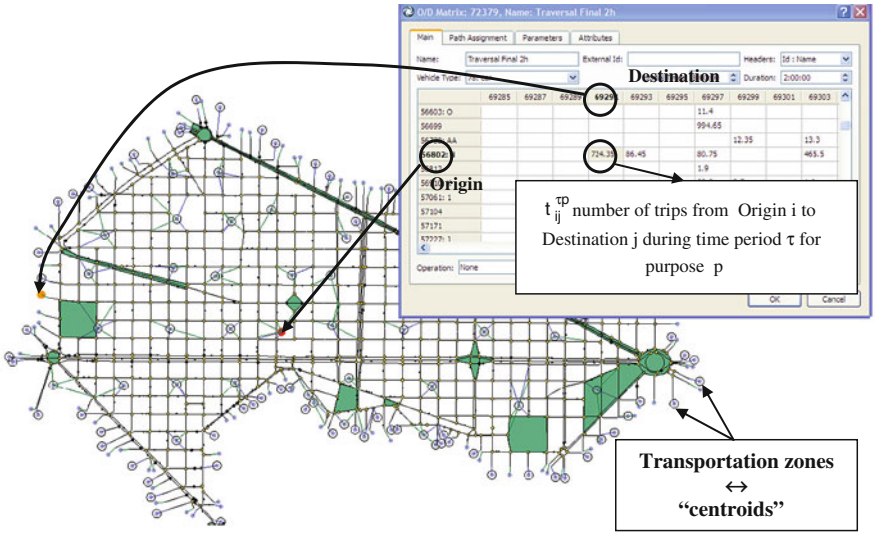


Fig. 1.3 Scheme of the aggregated representation of the demand in terms of an origin-to-destination OD matrix and its associated centroids

modeling the demand is in terms of an aggregated representation by means of an origin–destination (OD) matrix. Figure 1.3 synthesizes that representation.

The geographic region spanned by the transportation network object of study is divided in terms of transportation zones, each one generating and attracting trips for a given purpose during a time period. These transportation zones will generate the flows of trips from origins to destinations along the available paths on the network. Origins and destinations are usually modeled in terms of dummy nodes, or "centroids," in the network. The demand is then modeled in terms of a matrix whose rows are origins and whose columns are destinations, the entries $t_{ij}^{\tau p}$ of the matrix representing the number of trips from origin i to destination j during time period τ for the purpose p.

The right-hand part of Fig. 1.2 schematizes the system component. The assumption here is that transport authorities have their own assessment on how the system is performing. This is usually one of the functions of the equipment on the road: loop detectors measuring traffic variables (i.e., volumes, occupancies, and speeds); TV cameras, either monitoring the traffic or providing detector data through image-processing systems; and any other of the available technologies for collecting traffic data that, when suitably processed, provide authorities with additional evidence for estimating the state of the network. It is also assumed that authorities have specific objectives whose goal is to avoid or alleviate, if possible, conflicts which may escalate in the network, that is, manage the network in the most efficient way possible in order to minimize delays, travel times, manage congestion, etc. This is achieved by means of traffic management schemes, traffic control policies, and other management strategies. The set of control and management strategies and the conditions for

their use determines the performance of the transportation system or, in other terms, the transportation system's capacity for allocating the demand.

Consequently, along with this interpretation comes another main component of the transportation model: the model of the network and its conditions of use in terms of traffic management schemes and traffic control policies.

The network is modeled in a way which depends on the objectives of the transportation system analysis and other modeling approaches that will be used accordingly. This is a nice example of the former assertion that there not only is a single system model but also are numerous models that are suitable for answering the questions that observers – transport analysts in our case – ask about the system's behavior and performance under different conditions, as well as other issues.

With most of the available transportation analysis software, the current trends in transportation network modeling employ graphic editors to translate the digital map into a network model, which is currently available in the geographic information systems. And depending on the type of analysis, the required information is added to the geometry. Figure 1.4 depicts an example. On the left it shows a digital map translated into a graphical representation of the road network, $G=(N, A)$, whose nodes represent intersections and centroids and whose links represent the transportation infrastructure, roads or streets – depending on whether the model is of an interurban or an urban system.

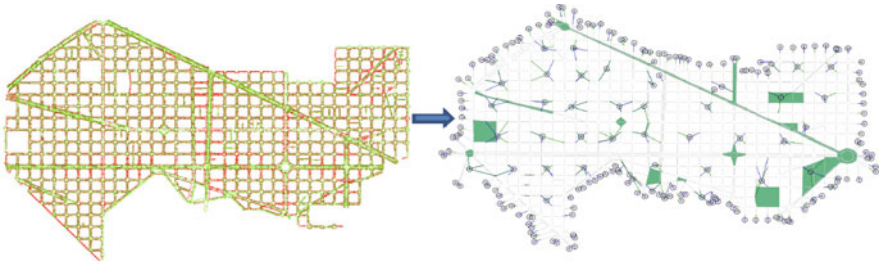


Fig. 1.4 From the digital map to the graphical representation of the road network

For those modeling approaches based on this type of network representation, other additional information is necessary. Most of the modeling approaches are explicitly or implicitly based on an object-entity approach which characterizes the entities of the system in terms of attributes. In the case of transportation modes, a system's entity could be a link representing a road section; its attributes can be the link capacity, number of lanes, transportation modes that can use each lane, volume–delay functions that compute the link travel time as a function of the link's traffic flow volume, speed–density relationships governing the dynamics of the traffic flow in the link, jam density, etc.

Some modeling approaches ask for a more detailed description of the network. Figure 1.5 depicts an example of an extended graphical representation of the traffic network on the left, in which – rather than being represented by a unique

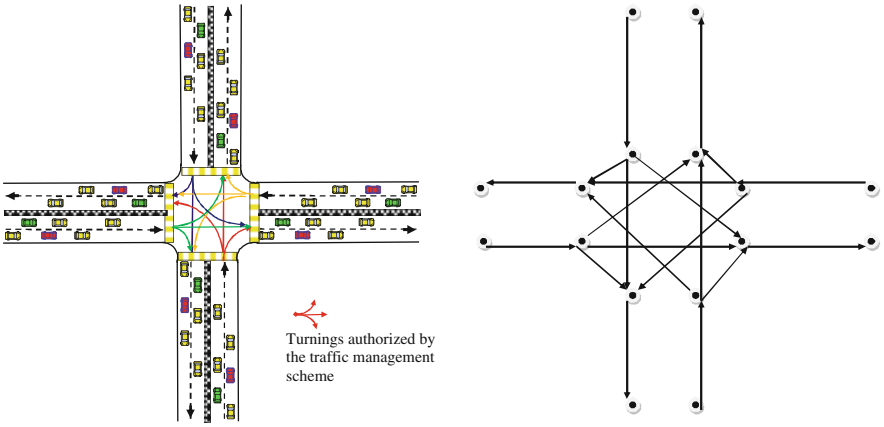


Fig. 1.5 Extended graphical representation of an intersection accounting for all allowed turnings

node – an intersection with a set of allowed turnings is split into a set of auxiliary nodes to account for all turnings.

This type of expansion is required when penalties or volume–delay functions are also associated with turnings or when the modeling approach also deals explicitly with traffic control settings.

In other modeling approaches, the network model requires a more detailed network representation and must explicitly take into account the network geometry as it is, reproducing it as accurately as possible, for example, as it appears in the left side of Fig. 1.5. The attributes in this case will be not only the number of lanes but also the lane width, the turning radius, speed limits at the links and turnings, the explicit specification of traffic control settings, variable message panels, and perhaps other traffic-related objects like traffic detectors (locations and functions), depending on the model objectives. Other attributes are also possible.

In summary the component of the model of the transport system corresponding to the network model consists of the following:

- Different degrees of details of the network geometry ranging from a simple graphical representation to a very detailed representation of the geometry, depending on the modeling approach and the model purposes
- An explicit description of the traffic control settings
- Possibly other objects (detectors, variable message panels, etc.), depending on the model objectives

We have so far described generically two major components of the transportation system following the conceptual approach of the diagram in Fig. 1.2, the travel demand and the capacity of the transportation network. But to complete our model-building process according to the proposed systems approach, we must be able to formalize the relationships between the capacity and the demand. To model

this interaction, the main underlying hypothesis is that travelers travel from origin to destinations in the network along the available routes connecting them, which involves modeling how travelers chose their routes through the network. The modeling hypothesis that supports the main transportation models is based on the concept of user equilibrium, which assumes that travelers try to minimize their individual travel times, that is, travelers chose the routes that they perceive as the shortest under the prevailing traffic conditions. This modeling hypothesis is formulated in terms of Wardrop's first principle (1952): *The journey times on all the routes actually used are equal, and less than those which would be experienced by a single vehicle on any unused route.*

Traffic assignment is the process of determining how demand traffic, usually defined in terms of an origin–destination matrix, is loaded onto the network, and it provides the means for computing traffic flows on the network links. Traffic assignment models based on Wardrop's principle are known as user equilibrium models (Sheffi, 1985; Florian and Hearn, 1995). This modeling hypothesis, implemented for traffic demands and average flows not depending on the time of day, has supported the traditional transport-planning models used in practice for strategic planning analysis. But our objective is for a more detailed modeling of traffic phenomena dealing explicitly with their time dependencies; therefore the modeling hypothesis that we need to explicitly account for the interactions between traffic demand and the capacity of the transportation system must

- support a route choice mechanism that provides a procedure for loading a time-dependent demand onto the network and that explicitly deals with time dependencies of traffic flows on the network links, determining the paths that will be used and the proportion of the demand at each instant in time.
- be able to describe the traffic flow dynamics which explains these time dependencies, that is, a "network loading process" that describes how flows propagate with time through the network along the selected paths.

The general modeling principles described so far, when applied to traffic and transportation models, can be conceptually described in terms of the logic diagram of Fig. 1.6.

1.3 Algorithmic Framework for Dynamic Traffic Models: Dynamic Traffic Assignment and Dynamic User Equilibrium

The dynamic transportation models that have been described correspond to the dynamic traffic assignment (DTA) problem, an extension of the traffic assignment problem mentioned above able to determine the time variations in link or path flows and capable of describing how traffic flow patterns evolve in time and space in the network (Mahmassani, 2001).

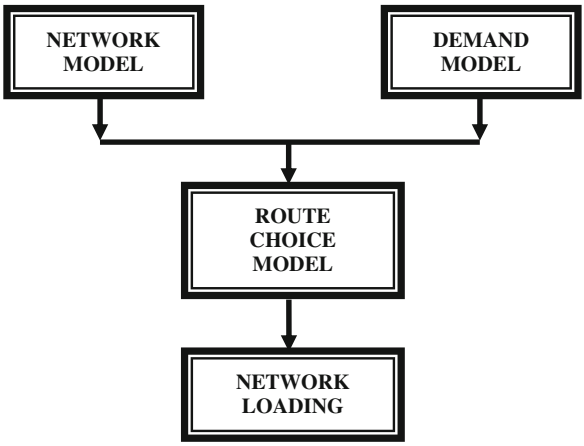


Fig. 1.6 Conceptual approach to a dynamic transportation model

For a DTA to become a dynamic user equilibrium (DUE), the behavioral assumptions on how travelers choose the routes have to be consistent with the dynamic user equilibrium principle. Ran and Boyce (1996) formulated the dynamic version of Wardrop's user equilibrium in the following terms: *If, for each OD pair at each instant of time, the actual travel times experienced by travelers departing at the same time are equal and minimal, the dynamic traffic flow over the network is in a travel time-based dynamic user equilibrium (DUE) state.* Friesz et al. (1993) show that the DUE approach can be implemented in terms of solving the following mathematical model

$$\begin{aligned}
&\left[\tau_{rsp}(t) - \theta_{rs}(t)\right] f_{rsp}(t) = 0, \qquad \forall p \in P_{rs}(t), \forall (r,s) \in \Im, t \in [0,T] \\
&\tau_{rsp}(t) - \theta_{rs}(t) \geq 0, \qquad \forall p \in P_{rs}(t), \forall (r,s) \in \Im, t \in [0,T] \\
&\tau_{rsp}(t), \theta_{rs}(t), f_{rsp}(t) \geq 0
\end{aligned} \tag{1.1}$$

and the flow balancing equation

$$\sum_{p \in P_{rs}(t)} f_{rsp}(t) = d_{rs}(t), \quad \forall (r,s) \in \Im, t \in [0,T] \tag{1.2}$$

where $f_{rsp}(t)$ is the flow on path p from origin r to destination s departing origin r at time interval t, $\tau_{rsp}(t)$ is the actual path cost from r to s on route p at time interval t, $\theta_{rs}(t)$ is the cost of the shortest path from r to s departing from origin r at time interval t, $P_{rs}(t)$ is the set of all available paths from r to s at time interval t, $\Im$ is the set of all origin–destination pairs (r, s) in the network, $d_{rs}(t)$ is the demand (number of trips) from r to s at time interval t, and T is the time horizon. It can be shown that this is equivalent to solving a finite-dimensional variational inequality problem which consists of finding a vector of path flows f^* such that

$$\left[f - f^*\right]^{\mathrm{T}} \tau \geq 0, \forall f \in \Theta$$
$$\Theta = \left\{ f_{rsp}(t) \left| \sum_{p \in P_{rs}(t)} f_{prs}(t) = d_{rs}(t), \forall (r,s) \in \Im, t \in [0,T], f_{rsp}(t) \geq 0 \right. \right\} \quad (1.3)$$

Wu et al. (1991, 1998a, b) prove that this is equivalent to solving the discretized variational inequality:

$$\sum_{t} \sum_{p \in \Re} \tau_{rsp}(t) \left[f_{rsp}(t) - f^*_{rsp}(t) \right] \geq 0, \qquad t = 0, 1, 2, ..., \frac{T}{\Delta t} \quad (1.4)$$

where $\Re = \bigcup_{(r,s \in \Im)} P_{rs}$ is the set of all available paths and Δt is the departure time interval.

To solve the dynamic traffic assignment model, Florian et al. (2001, 2002) propose an algorithmic framework consisting of two main components:

1. A method for determining the path-dependent flow rates on the paths within the network
2. A dynamic network loading method, which determines how these path flows give rise to time-dependent arc volumes, arc travel times, and path travel times

Various algorithmic schemes have been proposed for implementing this algorithmic framework into practice, ranging from purely analytical to heuristic approaches. In the former case, path flow rates and dynamic network loading are implemented analytically (Wu, 1991; Wu et al., 1998a, b; Xu et al., 1998, 1999). Heuristic approaches estimate path flow rates on the basis of stochastic algorithms, which are aimed first at emulating the user's route choice decision making and then emulating the dynamic network loading.

This network loading simulates the traffic flow dynamics or combines analytical and heuristic methods to numerically solve eq. (1.4) to get time-dependent flow rates that ensure a DUE solution. From there the network flows propagate through the simulated dynamic network loading. Figure 1.7 provides a scheme of this computational framework assuming various convergence criteria depending on the algorithmic approach.

However, it should be highlighted that not all computational implementations of this algorithmic framework provide DUE solutions. Route choice algorithms can be grouped into two classes: preventive (Papageorgiou, 1990), which implicitly assumes that traffic conditions in the network are predictable and decision makers are aware of these conditions, e.g., by historical experience, and reactive, which assumes that traffic conditions in the network are not predictable due to incidents, variability of demand, stochasticity of the traffic system, and so on. But users have real-time information on the current traffic conditions, e.g., the travel times that they had experienced, and can make route decisions while en route. Friesz et al. (1993)

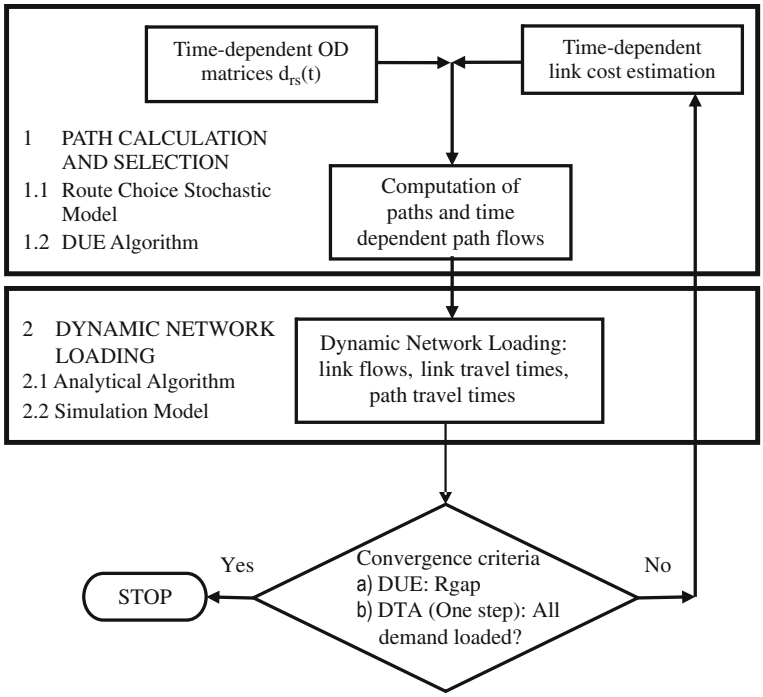


Fig. 1.7 Computational framework for dynamic transportation models (adapted from Florian et al., 2001)

prove that DUE solutions are reached through the implementations of the preventive route choice mechanism, combining experienced travel times with conjectures to forecast temporal variations in flows and travel cost. A variety of algorithms have been proposed for explicitly solving the set of variational inequalities (1.4) to provide DUE solutions: from projection algorithms (Wu et al., 1991, 1998a, b; Florian et al., 2001) or methods of alternating directions (Lo and Szeto, 2002) to various versions of the method of successive averages (MSA) (Tong and Wong, 2000; Varia and Dhingra, 2004; Florian et al., 2002; Mahut et al., 2003a, b, 2004).

Other proposals which can be considered a dynamic traffic assignment from the reactive implementation of the route choice decision making are those that model the process from the point of view of the discrete choice theory (Ben-Akiva and Lerman, 1985). This approach considers that $P_{rs}(t)$, the set of all available paths from r to s at time interval t, is a finite choice set of alternatives, each one with a perceived utility for the decision maker, i.e., the traveler. Examples of this could be travel time or travel cost at time t. The utility for each alternative k can be considered a random variable consisting of a systematic, deterministic component $C_k[v(t)]$; the measured utility (where $v(t)$ is the vector of the values at time t of the variables on which this utility depends); and an additive random error $\varepsilon_k(v)$, representing the perception error due to the lack of perfect information. Then the perceived utility of alternative k (path k) at time t is

$$U_k(t) = -\theta C_k[v(t)] + \varepsilon_k(v), \forall k \in P_{rs}(t)$$

and θ is a positive parameter when $C_k[v(t)]$ is the expected value of a negative utility, e.g., the expected travel cost or travel time. Assuming that the random terms satisfy the condition that their expected values are $E[\varepsilon_k(v)] = 0$, $\forall k$ and are independent, identically distributed random Gumbel variates, it can be proven that the choice probability $P_k(t)$ of alternative k (path k) at time t is given by the logit function

$$P_k(t) = \frac{e^{-C_k[v(t)]}}{\sum_{j \in P_{rs}(t)} e^{-C_j[v(t)]}}$$

A well-known drawback of logit choice functions that are used to select paths is that they do not distinguish between overlapping paths in order to overcome the undesired side effects of wrong choices. Some researchers (Cascetta et al., 1996; Ben-Akiva and Bierlaire, 1999) have proposed a modified logit that adds to the utility a penalty term which is a function of the degree of overlapping between alternative paths. In this model, the choice probability P_k of each alternative path k belonging to $P_{rs}(t)$, the set of all available paths from r to s at time interval t, is defined as

$$P_k(t) = \frac{e^{-\theta\{C_k[v(t)]+CF_k\}}}{\sum_{j \in P_{rs}(t)} e^{-\theta\{C_j[v(t)]+CF_j\}}}$$

where $C_k[v(t)]$ is the expected value of the perceived utility for alternative path k at time t, i.e., the opposite of the path cost, and θ is the scale factor, as in the case of the logit model. The term CF_k, denoted as "commonality factor" of path k, is directly proportional to the degree that path k overlaps with other alternative paths. Thus, highly overlapped paths have a larger CF factor and therefore smaller utility with respect to similar paths. An example of commonality factor CF_k (Cascetta et al., 1996) could be

$$CF_k = \beta \ln \sum_{j \in P_{rs}(t)} \left(\frac{L_{jk}}{L_j^{1/2} L_k^{1/2}} \right)^{\gamma}$$

where L_{jk} is the length of arcs common to paths j and k, while L_j and L_k are the length of paths j and k, respectively. Depending on the two factor parameters β and γ, a greater or lesser weighting is given to the "commonality factor."

In the dynamic network loading, also known as dynamic network flow propagation (Cascetta, 2001), "models simulate how the time-varying continuous path flows propagate through the network inducing time-varying inflows, outflows and link occupancies." Let us begin by defining what we understand by simulation. Law

and Kelton (1991) define simulation as the set of techniques that employ computers to imitate – or *simulate* – the operations of various kinds of real-world facilities or processes. The facility or process of interest is usually called a *system*, and in order to study it scientifically we often have to make a set of assumptions about how it works. May (1990) defines simulation as a *numerical technique for conducting experiments on a digital computer*, which may include stochastic characteristics, be microscopic or macroscopic in nature, and involve *mathematical models* that describe the behavior of a system over extended periods of real time. Simulation can thus be seen as an alternative to analytical models consisting of a technique that imitates on a computer the operation of a real-world system as it evolves over time. In what follows we will address how the dynamics of traffic flows can be simulated.

A key aspect in a simulation process concerns how the simulation model evolves with time. There are two methodological approaches for dealing with time in simulation: asynchronous and synchronous timing. Synchronous timing in simulation corresponds to time-oriented simulations in which time in the model is advanced at an appropriately chosen unit time Δt, the simulation step. Asynchronous or event-based simulations are those in which time advances in variable amounts that correspond to the instants in time at which occur events that change the model state. With few exceptions – which will be discussed later on – the main approaches to traffic simulation are based on a synchronous advance of time at predefined simulation steps.

1.4 Principles of Traffic Flow Modeling

Modeling the dynamics of traffic flows to simulate their temporal propagation through traffic networks is also a nice illustration of Minsky's statement that a system can be modeled in different ways according to various approaches depending on the modeler's purposes. Traffic flows can be modeled macroscopically from an aggregated point of view based on a hydrodynamic analogy by regarding traffic flows as a particular fluid process whose state is characterized by aggregate macroscopic variables: *density, volume, and speed*. But they can also be modeled microscopically, that is, from a fully disaggregated point of view aimed at describing the fluid process from the dynamics of the individual particles (the vehicles) that compose it. See, for instance, Chapters 6 and 7 of Gerlough and Huber (1975). Mesoscopic models represent a third intermediate modeling alternative based on a simplification of vehicular dynamics.

1.4.1 Macroscopic Modeling of Traffic Flows

The macroscopic modeling of traffic flows is usually based on the continuum traffic flow theory whose objective is the description of the time–space evolution of the variables characterizing the macroscopic flows: volume $q(x, t)$, speed $u(x, t)$, and density $k(x, t)$ which we assume are defined at every instant in time t and every point

in space x. The main equation formally representing this theory is the conservation equation (Gerlough and Huber, 1975; Kühne et al., 1992):

$$\frac{\partial q}{\partial x} + \frac{\partial k}{\partial t} = 0 \tag{1.5}$$

This is also known as the continuity equation. Similar to the continuity equation in hydrodynamics, it formally represents the assumption that, between two counting stations in a motorway section without entrances and exits, the number of vehicles is conserved. This equation is complemented by the fundamental relationship

$$q(x,t) = k(x,t)u(x,t) \tag{1.6}$$

To solve eq. (1.5), we need an additional equation which is usually based on the hypothesis that flow q is a function of density $q{=}q(k)$ or equivalently that speed is also a function of density $u{=}u(k)$, an assumption that holds up only in equilibrium conditions. The continuity equation (1.5) can be enhanced by adding a generation term $g(x, t)$ that represents the number of vehicles entering or leaving the traffic flow in a freeway with entries/exits:

$$\frac{\partial q}{\partial x} + \frac{\partial k}{\partial t} = g(x,t) \tag{1.7}$$

The speed–density relationship $u{=}\,u(k)$ must be provided by a theoretical or an empirical u–k model equation of state, which can take the general form (May and Keller, 1967)

$$u = u_{\mathrm{f}}\left[1 - \left(\frac{k}{k_{\mathrm{jam}}}\right)^{\alpha}\right]^{\beta} \tag{1.8}$$

where u_{f} is the free-flow speed and k_{jam} is the jam density. Since the simple continuum model does not consider acceleration and inertia effects, it does not faithfully describe non-equilibrium traffic flow dynamics. Payne (1971, 1979) suggested an improved model by replacing eq. (1.8) with a second partial differential equation corresponding to the momentum equation in fluid dynamics:

$$\frac{\partial k}{\partial t} + u\frac{\partial q}{\partial x} = \frac{1}{T}\left[u_{\mathrm{e}}(k) - u\right] - \frac{\nu}{k}\frac{\partial k}{\partial x}$$

where T is the relaxation time; ν is the anticipation parameter; the first term on the right-hand side is the relaxation to equilibrium, that is, the effects of drivers adjusting their speeds to the equilibrium speed–density relationship; and the second term represents the anticipation, that is, the effect of drivers reacting to downstream traffic conditions. Payne's model provided good results under certain traffic conditions but was found to lack accuracy under dense traffic conditions near on-ramps and/or lane drops.

A number of extensions (Papageorgiou et al., 1989, 1990a; 1990b; Ross, 1988; Michalopoulos et al., 1991; Kühne, 1989; Papageorgiou and Schmidt, 1991) contributed to an accuracy improvement of Payne's model. Most of these extensions include a relaxation term that represents the traffic flow tendency to adjust speeds due to changes in free-flow speeds along the roadway. This is an empirically estimated traffic friction term that models traffic friction at freeway ramp junctions due to ramp flows as a function of a friction parameter which depends on the ramp volumes entering or leaving. It is also an anticipation term that represents the effect of drivers reacting to downstream traffic conditions.

To numerically integrate these equations, each traffic model of the road section (space dimension) is discretized in time and space (Messmer and Papageorgiou, 1990; Papageorgiou et al., 1989, 1990a; Chronopoulos et al., 1992; Michalopoulos et al., 1991). Macroscopic traffic simulation models belong to this type of synchronous simulation approaches. Numerical methods, which are used in computational fluid dynamics, can be applied to solve these equations (Hirsch, 1988).

To account for dynamic effects of flow behavior, they also require the definition of time-dependent mobility patterns, that is, a time-sliced definition of the input flow into the input section of the model, as, for example, the q_{it} input flow through the ith entry ramp during time interval t, and the output flow at the exit ramps, such as the q_{jt} flow through the exit ramp j at time interval t. This last amount is often expressed in terms of the percentage of flow through the main section leaving at exit ramp j.

Numerical computation of k, u, and q proceeds by discretizing the roadway under consideration into small segments Δx and updating the values of these traffic flow variables on each node of the discretized network at consecutive time increments Δt (Michalopoulos, 1988; Papageorgiou and Schmidt, 1991). Space discretization of a simple traffic link is presented in Fig. 1.8.

Density on any node j, except those on the boundary, at the next time step n+1 is computed from density in the immediately adjacent links (both upstream and downstream j–1 and j+1, respectively) at the current time step n according to the relationship

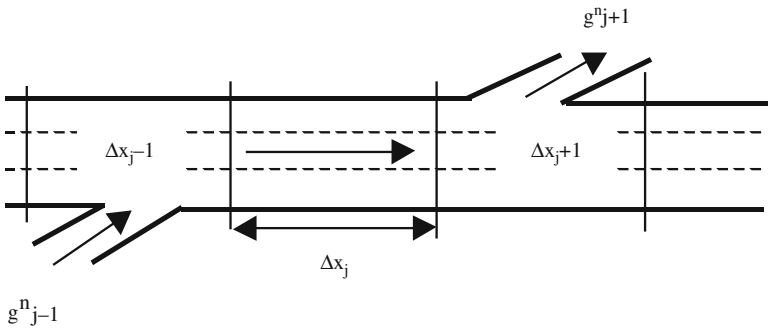


Fig. 1.8 Space discretization of a simple link

$$k_j^{n+1} = \frac{1}{2}\left(k_{j+1}^n + k_{j-1}^n\right) - \frac{\Delta t}{2\Delta x}\left(q_{j+1}^n - q_{j-1}^n\right) + \frac{\Delta t}{2\Delta x}\left(g_{j+1}^n + g_{j-1}^n\right)$$

in which

k_j^n, q_j^n is the density and flow rate on node j at $t = t_0 + n\Delta t$
t_0 is the initial time
Δt, Δx are the time and space increments, respectively, such that $\Delta x/\Delta t >$ free-flow speed
g_j^n is the generation (dissipation) rate on node j at $t = t_0 + n\Delta t$; if no sinks or sources exist, $g_j^n = 0$

Once the density is determined, the speed at $t+\Delta t$ at $n+1$ is obtained from the equilibrium speed relationship $u_e(k)$, i.e., $u_j^{n+1} = u_e\left(k_j^{n+1}\right)$, as for instance, the Greenshields linear model (Greenshields, 1934):

$$u_j^{n+1} = u_f\left(1 - \frac{k_j^{n+1}}{k_{jam}}\right)$$

There are other more advanced models, where u_f is the free-flow speed and k_{jam} is the jam density. It should be noted that this equation is applicable for any speed–density model including discontinuous ones; if an analytical expression is not available, then u can be easily obtained numerically from the u–k curve. Finally, flow at $t+\Delta t$ is obtained from the fundamental relationship:

$$q_j^{n+1} = k_j^{n+1} u_j^{n+1}$$

In which the values of k and u are first obtained from the previous equations. Measures of effectiveness such us delays, stops, total travel, etc., can be derived from k, u and q.

1.4.2 Microscopic Modeling of Traffic Flows

Microscopic modeling of traffic flows is based on the description of the motion of each individual vehicle composing the traffic stream. This implies modeling the actions – e.g., acceleration, decelerations, and lane changes – of each driver in response to the surrounding traffic. According to May (1990), theories describing how one vehicle follows another vehicle were developed primarily in the 1950s and 1960s, after the pioneering development of car-following theories by Reuschel (1950a, b) and Pipes (1953). Pipes' work, based on the concept of distance headway, characterizes the motion of vehicles in the traffic stream as following rules suggested in the California Motor Vehicle Code, namely "A good rule for following another vehicle at a safe distance is to allow yourself at least the length of a car between your vehicle and the vehicle ahead for every ten miles per hour of speed at which you are traveling." Pipes' car-following theory leads to a minimum safe distance headway that increases linearly with speed, a result that – considering the simplicity of the model – agrees acceptably well with the field calibration. Much

more extensive research was undertaken in the late 1950s by the General Motors Group based on comprehensive field experiments and the development of the mathematical theory bridging micro and macro theories of traffic flows. This research led to the formulation of the car-following models as a form of stimulus-response equation (Gerlough and Huber, 1975), where the response is the reaction of a driver to the motion of the vehicle immediately preceding him in the traffic stream. The response is always to accelerate or decelerate in proportion to the magnitude of the stimulus at time t and begins after a time lag T, the reaction time of the follower. The General Motors Group developed a series of models whose basis equation is of the form

$$\text{Response}\ (t+T) = \text{Sensitivity} \times \text{Stimulus}\ (t)$$

Models vary according to the various answers to the key questions:

- What is the nature of the driver's response?
- To what stimulus does he or she react and how do we measure his or her sensitivity?

The first and simplest model corresponds to the case when the response is represented by the acceleration or the deceleration of the follower driver and the stimulus is represented by the variation in the relative speeds. This simple model considers that the sensitivity is constant. If $x_n(t)$ and $x_{n+1}(t)$ are the positions of the leader and the follower, respectively, at time t, then the basic model is

$$\ddot{x}_{n+1}\,(t+T) = \lambda\left[\dot{x}_n\,(t) - \dot{x}_{n+1}\,(t)\right] \tag{1.9}$$

where the response is acceleration; deceleration, depending on the sign of the stimulus; positive if the relative speed is positive, that is, $\dot{x}_n\,(t) > \dot{x}_{n+1}\,(t)$ and negative if the relative speed is negative, that is, $\dot{x}_n\,(t) < \dot{x}_{n+1}\,(t)$; or no action when speeds are equal, $\dot{x}_n\,(t) = \dot{x}_{n+1}\,(t)$. This model is known as the linear car-following model because the response is directly proportional to the stimulus (Gerlough and Huber, 1975; Rothery, 1992). Gazis et al. (1959) analyzed the relationships between the linear car-following model and the stream macroscopic traffic models for steady-state flows. The integration of eq. (1.9) gives the speed of vehicle n+1, which can be interpreted in terms of the traffic stream velocity. In order to solve the resulting equation to calculate the integration constraint, taking into account that density k is the reciprocal of s, the average spacing between vehicles $s = x_n - x_{n+1}$ (see Gerlough and Huber, 1975 for details), the speed–density relation derived from eq. (1.9) is

$$u = \lambda\left[\frac{1}{k} - \frac{1}{k_{\text{jam}}}\right]$$

where k_{jam} is the jam density and, taking into account the fundamental relationship q=ku, the resulting steady-state equation is

$$q = ku = \lambda \left[1 - \frac{k}{k_{\text{jam}}} \right]$$

This is inconsistent with the observed traffic stream data (Gerlough and Huber, 1975; Rothery, 1992). To overcome this inconsistency, Gazis et al. (1959) proposed that the sensitivity constant should be inversely proportional to the headway, resulting in the model

$$\lambda = \frac{c_1}{x_n(t) - x_{n+1}(t)} \Rightarrow \ddot{x}_{n+1}(t+T) = \frac{c_1}{x_n(t) - x_{n+1}(t)} \left[\dot{x}_n(t) - \dot{x}_{n+1}(t) \right] \tag{1.10}$$

By integrating eq. (1.10) now, calculating the integration constraint in a similar way, and replacing the values for known steady flow conditions, the speed–density relationship is

$$u = c_1 \ln \left(\frac{k_{\text{jam}}}{k} \right)$$

Defining q_{m} as the maximum flow and k_{m} as the density at maximum flow, it can be shown that $c_1 = u_{\text{m}}$, the speed at maximum flow. From the relationship $q = ku$, the resulting steady-state equation is

$$q = k u_{\text{m}} \ln \left(\frac{k_{\text{jam}}}{k} \right)$$

The steady-state equation proposed by Greenberg (1959) was confirmed by experimental observations of flow, density, and velocity. This result has a twofold value: on the one hand, it proves consistency in the car-following theory and on the other hand, it will become a test for validating microscopic models through their capability in reproducing observed macroscopic results.

A short time later, Edie (1960) suggested a variation aimed at making the model more accurate for less-than-optimal traffic densities. The modified theory for non-congested traffic helps to describe the sudden changes of state occurring in a traffic stream in the transition from relatively free-flowing conditions to a crawling stop-and-go condition and back. Eddie's refinement assumes that the sensitivity of a driver varies with his absolute velocity, i.e., the faster the driving, the greater the sensitivity. What is more, inversely to the square of the headway, this means that when the leading vehicle is closer, the sensitivity to the absolute speed is greater. The model then becomes

$$\lambda = c_2 \frac{\dot{x}_{n+1}(t)}{\left[x_n(t) - x_{n+1}(t) \right]^2} \Rightarrow \ddot{x}_{n+1}(t+T) = c_2 \frac{\dot{x}_{n+1}(t)}{\left[x_n(t) - x_{n+1}(t) \right]^2} \left[\dot{x}_n(t) - \dot{x}_{n+1}(t) \right]$$

Edies's formulation was shown to be better at low flows due to its ability to predict a finite speed when density approaches zero. This result was the first to propose that two separate relationships could be used to model traffic flows: one for congested and another for uncongested conditions. Further research by Gazis,

Herman and Rothery (1961) emphasized the steady-state flow conditions resulting from various microscopic theories of traffic flow and proved that some of the proposed models could be considered as particular cases of the general model proposed by the General Motors Group:

$$\lambda = c\frac{\dot{x}_{n+1}^{m}(t)}{\left[x_n(t) - x_{n+1}(t)\right]^{l}} \Rightarrow \ddot{x}_{n+1}(t+T) = c\frac{\dot{x}_{n+1}^{m}(t)}{\left[x_n(t) - x_{n+1}(t)\right]^{l}}\left[\dot{x}_n(t) - \dot{x}_{n+1}(t)\right]$$

Empirical research based on available data sets tried to find the most suitable values for parameters l and m. May and Keller (1967) found them to be m=1 and l=3. Better values (m=0.8 and l=2.8) can be found if non-integer values are allowed.

A generalized version of this model has been proposed by Ahmed (1999) and assumes an acceleration rate given by the following equation:

$$\ddot{x}_{n+1}(t) = \alpha^{\pm}\frac{\dot{x}_{n+1}^{\beta\pm}(t)}{g_{n+1}^{\gamma\pm}(t)}\left(\dot{x}_n(t) - \dot{x}_{n+1}(t)\right) \tag{1.11}$$

where $\alpha^{\pm}$, $\beta^{\pm}$. and $\gamma^{\pm}$ are model parameters, α^{+}, β^{+}, γ^{+} are used for acceleration ($\dot{x}_{n+1}(t) \leq \dot{x}_n(t)$), α^{-}, β^{-}, γ^{-} are used for deceleration ($\dot{x}_{n+1}(t) > \dot{x}_n(t)$), l_n is the vehicle's length, and $g_{n+1} = x_{n+1} - x_n - l_n$ represents the gap distance from the leading vehicle. This generalization assumes a different behavior of drivers of follower vehicles, depending on whether they are in acceleration or deceleration phase. Some other refinements have been proposed with various results. For further details, the reader is referred to the monographs by Gerlough and Huber (1975) and Rothery (1992).

Another modeling approach, as in Pipes' model, is the collision avoidance approach that assumes that a driver will place himself at a distance $\Delta(t)$ from the lead vehicle such that – in the event of an emergency stop by the leader – the follower will come to rest without striking the lead vehicle. The safe deceleration to stop diagram in Fig. 1.9 (Gerlough and Huber, 1975; Mahut, 1999) illustrates how it works.

The time–space diagram shows the position of the leader vehicle n at time t when it starts to brake until it comes to a complete stop at time τ; the position of the follower vehicle n+1 at time t+T when it starts to brake with a delay T; the reaction time after the follower perceives that the leader is breaking; and the position when he safely comes to a stop. If Sd_n is the stopping distance for vehicle n – that is, the distance required by vehicle n to stop when traveling with speed $v_n(t) = \dot{x}_n(t)$ at time t and braking with deceleration $b_n(t)$, then Sd_n is given by $\mathrm{Sd}_n = f_n[v_n(t), b_n(t)]$, a function of $v_n(t)$ and b. Similarly the stopping distance for the follower vehicle n+1 is given by the function $\mathrm{Sd}_{n+1} = f_{n+1}[v_{n+1}(t), b_{n+1}(t)]$ of the current speed of the follower $v_{n+1}(t) = \dot{x}_{n+1}(t)$ and its braking capabilities $b_{n+1}(t)$. The desired spacing $s(t) = x_n(t) - x_{n+1}(t)$ at time t for a safe deceleration to stop is then given by the relationship

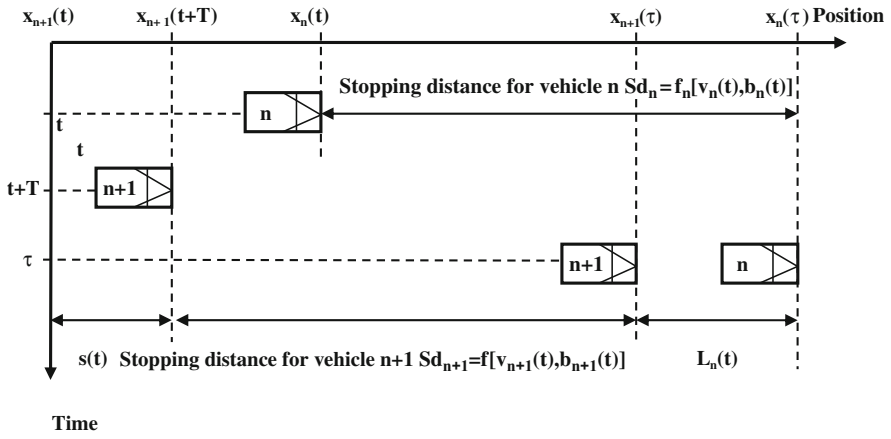


Fig. 1.9 Safe deceleration to stop diagram

$$s(t) = x_n(t) - x_{n+1}(t) = T\dot{x}_{n+1}(t) + \mathrm{Sd}_{n+1}\left[v_{n+1}(t+T), b_{n+1}(t+T)\right] + L_n(t) - \mathrm{Sd}_n\left[v_n(t), b_n(t)\right] \tag{1.12}$$

where $T\dot{x}_{n+1}(t)$ is the distance traveled by the follower during the reaction time and $L_n(t)$ is the distance from front bumper to front bumper at rest. Assuming steady-state conditions (Gerlough and Huber, 1975) in which $b_{n+1}(t) = b_n(t)$, the speeds are equal and $\mathrm{Sd}_{n+1}(t) = \mathrm{Sd}_n(t)$, which results in

$$x_n(t) - x_{n+1}(t) = T\dot{x}_{n+1}(t+T) + L_n(t)$$

and differentiating with respect to t

$$\dot{x}_n(t) - \dot{x}_{n+1}(t) = T\ddot{x}_{n+1}(t+T) \Rightarrow \ddot{x}_{n+1}(t+T) = T^{-1}\left[\dot{x}_n(t) - \dot{x}_{n+1}(t)\right]$$

This is of the form Response $(t + T)$ = Sensitivity $\times$ Stimuli(t), which was previously proposed.

In all previous discussions, it has been assumed that the follower driver will adjust his reactions to a change in velocity of the leader accelerating and decelerating at the same rate for a given perception of the stimulus. However, it is obvious that the deceleration capabilities are usually greater than the acceleration capabilities. This was already observed by Herman and Rothery (1965), who proposed to modify the linear model (1.9) so that

$$\ddot{x}_{n+1}(t+T) = \lambda_+\left[\dot{x}_n(t) - \dot{x}_{n+1}(t)\right] \qquad \text{for positive relative velocity}$$

$$\ddot{x}_{n+1}(t+T) = \lambda_-\left[\dot{x}_n(t) - \dot{x}_{n+1}(t)\right] \qquad \text{for negative relative velocity}$$

This modification is already included in the general model (1.11).

Taking into account these different behaviors and the fact that from the behavioral point of view, other factors, for example, the target or the desired speed of a driver, should also be taken into account, Gipps (1981) developed an empirical (behavioral instead of "response to a stimulus") model consisting of two components: acceleration and deceleration, defined on the basis of variables that can be measured. The first represents the intention of a vehicle to achieve a certain desired speed, while the second reproduces the limitations imposed by the preceding vehicle when trying to drive at the desired speed. The deceleration component can be derived from the safe deceleration to stop relationship (1.12), which can be rewritten as

$$x_n(t) + \mathrm{Sd}_n[v_n(t), b_n(t)] - L_n(\tau) \geq x_{n+1}(t) + T\dot{x}_{n+1}(t) + \mathrm{Sd}_{n+1} \times \left[v_{n+1}(t+T), b_{n+1}(t+T)\right] \tag{1.13}$$

This can be interpreted in terms of a safety constraint which becomes active when satisfied as equality. Assuming steady-state conditions and that $L_n(\tau) = L_n$, b_n, and b_{n+1} are constants for a time period, the distance to stop for the leader vehicle n when traveling with speed $v_n(t) = \dot{x}_n(t)$ and braking with negative constant acceleration b_n is

$$\mathrm{Sd}_n = -\frac{\dot{x}_n^2(t)}{2b_n}$$

And similarly the distance to stop for the follower vehicle n + 1 is

$$\mathrm{Sd}_{n+1} = -\frac{\dot{x}_{n+1}^2(t+T)}{2b_{n+1}}$$

And taking into account that the follower will not start to brake until time $t + T$, the safety constraint (1.13) can be rewritten as

$$x_n(t) - \frac{\dot{x}_n^2(t)}{2b_n} - L_n(\tau) \geq x_{n+1}(t) + \left[\dot{x}_{n+1}(t) + \dot{x}_{n+1}(t+T)\right]\frac{T}{2} - \frac{\dot{x}_{n+1}^2(t+T)}{2b_{n+1}} \tag{1.14}$$

At this point, Gipps makes the observation that this relationship gives the driver no margin for error and he proposes a further safety margin for a delay θ when traveling at speed $\dot{x}_{n+1}(t+T)$, T+θ would then be the safely reaction time, whereas (1.14) then becomes

$$x_n(t) - \frac{\dot{x}_n^2(t)}{2b_n} - L_n(\tau) \geq x_{n+1}(t) + \left[\dot{x}_{n+1}(t) + \dot{x}_{n+1}(t+T)\right]\frac{T}{2} + \dot{x}_{n+1}(t+T)\,\theta - \frac{\dot{x}_{n+1}^2(t+T)}{2b_{n+1}} \tag{1.15}$$

Gipps shows that the value of θ is $T/2$ and underlines that all parameters in eq. (1.15) can be estimated by the follower driver by direct observation, with the exception of the maximum deceleration of the leader b_n, which can only be guessed.

Then if $\hat{b}$ is the estimate from the results of eq. (1.15), the maximum allowed speed for the follower is bounded by the safety constraint:

$$v_{n+1}^{d}(t+T) = \dot{x}_{n+1}(t+T) \leq b_{n+1}T + \sqrt{b_{n+1}^2T^2 - b_{n+1}\left[2\left(x_n(t) - x_{n+1}(t) - L_n\right)\right] - \dot{x}_{n+1}(t)\,T - \frac{\dot{x}_n^2(t)}{\hat{b}}} \tag{1.16}$$

This is the deceleration component of Gipps' (1981) car-following model. The acceleration component, corresponding to traffic conditions when the safety constraint is not active, was empirically estimated by fitting the envelope of a set of speeds and accelerations were measured by an instrumented car; it is given as

$$v_{n+1}^{a}(t+T) = \dot{x}_{n+1}(t+T) \leq \dot{x}_{n+1}(t) + 2.5a_{n+1}T\left(1 - \frac{\dot{x}_{n+1}(t)}{V_{n+1}}\right) \times \sqrt{0.025 + \frac{\dot{x}_{n+1}(t)}{V_{n+1}}} \tag{1.17}$$

where a_{n+1} is the maximum acceleration that the driver of the follower vehicle wishes to apply and V_{n+1} his or her desired or target travel speed. The final speed $v_{n+1}(t+T)$ of the follower vehicle n+1 at the end of the time interval $t + T$ is the minimum of these speeds:

$$v_{n+1}(t+T) = \text{Min}\left\{v_{n+1}^{a}(t+T), v_{n+1}^{d}(t+T)\right\} \tag{1.18}$$

Some researchers (Parker, 1996; Chen et al., 1995) found that the assumption that drivers follow a leading vehicle at a safe distance is frequently not respected. Hidas (1998) proposed a model based on these findings; the basic assumption of this model is based on the fact that drivers of a follower vehicle n+1 tend to follow closer than a "safe distance" and therefore, when approaching and following a leader vehicle n at any time, they attempt to adjust their acceleration so as to reach a "desired spacing" after a time lag which takes τ seconds. That is

$$x_n(t+\tau) - x_{n+1}(t+\tau) = D_{n+1}(t+\tau)$$

The main hypothesis is that the desired spacing is assumed to be a linear function of the desired speed:

$$D_{n+1}(t+\tau) = \alpha\dot{x}_{n+1}(t+\tau) + \beta$$

where α and β are constants and $\dot{x}_{n+1}(t+\tau)$ is, as before, the speed of the follower vehicle n+1 at time t+τ. An advantage of this model is that it does not depend on the behavioral aspects associated with the reaction times. Assuming constant

accelerations a_n and a_{n+1} for leader and follower, respectively, during time lag τ, Hidas (1998) proves that the follower acceleration a_{n+1} is given as

$$a_{n+1} = \frac{\tau\left[\dot{x}_n(t) - \dot{x}_{n+1}(t)\right]}{\alpha\tau + \frac{1}{2}\tau^2} + \frac{\left[x_n(t) - x_{n+1}(t) - \alpha\dot{x}_{n+1}(t) - \beta\right]}{\alpha\tau + \frac{1}{2}\tau^2} + \frac{\frac{1}{2}\tau^2 a_n}{\alpha\tau + \frac{1}{2}\tau^2} \tag{1.19}$$

This also assumes that when the follower driver approaches a slower leader, he or she attempts to set his or her acceleration to achieve the same speed as that of the leader at the end of the deceleration process. That is

$$\dot{x}_{n+1}(t) + a_{n+1}\tau = \dot{x}_n(t) + a_n\tau$$

The time lag τ can be calculated as

$$\tau = \frac{x_n(t) - x_{n+1}(t) - \alpha\dot{x}_n(t) - \beta}{\alpha a_n - 0.5\left[\dot{x}_n(t) - \dot{x}_{n+1}(t)\right]}$$

A different approach was taken by Wiedemann (1974) and Fellendorf (1994) in the mid-1970s to derive the so-called psycho-physical spacing models, whose description, based on Leutzbach (1988), is derived from two main assumptions:

1. At large spacing, the driver of a following vehicle is not influenced by the size of the speed difference, and
2. At small spacing, there are combinations of relative speeds and distance headways for which there is, as in 1, no response from the driver of the following vehicle because the relative motion is too small.

This implies the existence of perceptual thresholds such that only when they are reached will the driver of a following vehicle be able to perceive the change and subsequently be able to react. These thresholds graphically displayed in the $(\Delta x, \Delta\dot{x})$ space are represented by parabolas from which how the car following proceeds can be explained.

A vehicle with speed $\dot{x}_{n+1}$ larger than the speed $\dot{x}_n$ of the preceding vehicle will catch up with it at a constant relative speed $(\Delta\dot{x})$. Upon reaching the threshold, the driver reacts by reducing his speed. The relative motion with constant deceleration appears as a parabola in which the minimum speed lies on the (Δx) axis. The driver tries to decelerate so as to reach a point at which $(\Delta\dot{x} = 0)$. He is not able to do this accurately because he is neither able to perceive small differences nor able to control his speed sufficiently well. The result is that the spacing will again increase. When the driver first reaches the opposite threshold, he accelerates and again tries to achieve the desired spacing (for further details, see Chapter 2) If one assumes that the relationship of the perceptual thresholds for spacing is the same for both

positive and negative changes in relative speed, then the resulting spacing behavior resembles a symmetrical pendulum about its equilibrium point.

It can be shown (Leutzbach, 1988) that the described behavior corresponds to a particular case of the general car-following model:

$$\ddot{x}_{n+1}(t+T) = c\frac{\dot{x}_{n+1}^{m}(t)}{\left[x_n(t) - x_{n+1}(t)\right]^{l}}\left[\dot{x}_n(t) - \dot{x}_{n+1}(t)\right]$$

with m=0 and l=2.

In 2002 Newell proposed a new approach to modeling car following based on the analysis of time–space trajectories, assuming that the time–space trajectories for the nth and n+1th vehicles are essentially the same with a time–space translation. This approach assumes a reformulation of the goal of car-following models in terms of describing the dependencies between the trajectory x_{n+1} of the follower vehicle n+1 and the trajectory $x_n(t)$ of the leading vehicle n. The approach assumes that the spacing $s_n(t) = x_n(t) - x_{n+1}(t)$ depends on the average velocity v of both vehicles, that is, there is a relationship between $s_n(t)$ and v, e.g., for large v, the spacing will be larger. It also assumes that each driver has a "desired or target speed" V_n. Figure 1.10 shows the change in trajectories when the leader vehicle n travels at a constant average speed v for some time period before changing to a new constant speed v' and the follower follows the leader.

From the time displacement τ_{n+1} and the space displacement d_{n+1}, the following relationships follow:

$$d_{n+1} + v\tau_{n+1} = s_n \qquad \text{and} \qquad d_{n+1} + v'\tau_{n+1} = s'_n$$

where d_{n+1} and τ_{n+1} are independent of v. The piecewise linear approximation to trajectory $x_{n+1}(t)$ is the translation of the piecewise linear approximation to trajectory $x_n(t)$ by distance d_{n+1} and time τ_{n+1}. The car-following model approach is then based on an approximate calculation of the trajectory $x_{n+1}(t)$ of the follower vehicle

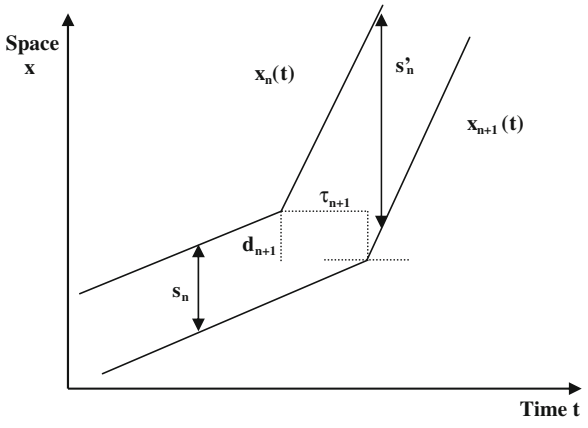


Fig. 1.10 Piecewise linear approximation to vehicle trajectories (Newell, 2002)

n+1 as the translation of the trajectory $x_n(t)$ of the leader vehicle n for the appropriate d_{n+1} and τ_{n+1}:

$$x_{n+1}(t + \tau_{n+1}) = x_n(t) + d_{n+1} \tag{1.20}$$

The value for τ_{n+1} derives from what the follower driver could consider to be a safe distance: that which allows him or her to respond comfortably to whatever changes the leader vehicle may do. For vehicles in which the velocity is smaller than V_k, $x_{n+1}(t + \tau_{n+1})$ can be written as

$$x_{n+1}(t + \tau_{n+1}) = x_{n+1}(t) + \tau_{n+1} v_{n+1}(t + T_{n+1}) \tag{1.21}$$

and approximated as

$$x_{n+1}(t + \tau_{n+1}) = x_{n+1}(t) + \tau_{n+1} v_{n+1}(t) + \tau_{n+1} T_{n+1} a_{n+1}(t) \tag{1.22}$$

where $v_{n+1}(t)$ is the velocity of vehicle n+1 at time t and a_{n+1}(t) is its acceleration. From eqs. (1.20) and (1.21)

$$v_{n+1}\,(t + T_{n+1}) = \frac{1}{\tau_{n+1}} \left[x_n\,(t) - x_{n+1}\,(t)\right] - \frac{d_{n+1}}{\tau_{n+1}} \tag{1.23}$$

and deriving with respect to time

$$a_{n+1}\,(t + T_{n+1}) = \frac{1}{\tau_{n+1}} \left[v_n\,(t) - v_{n+1}\,(t)\right] \tag{1.24}$$

From eqs. (1.20) and (1.24)

$$T_{n+1} a_{n+1}\,(t) = \frac{1}{\tau_{n+1}} \left[x_n\,(t) - x_{n+1}\,(t)\right] - \frac{d_{n+1}}{\tau_{n+1}} - v_{n+1}\,(t) \tag{1.25}$$

Newell's model can be interpreted by assuming that the follower driver chooses a velocity based on time spacing T_{n+1} and an acceleration based on the velocity difference at time t, which is proportional to his deviation from an equilibrium curve with a "relaxation time" T_{n+1}.

All car-following models summarily described so far depend on a number of parameters aimed at mimicking as closely as possible the way in which drivers of follower vehicles adjust their driving to that of leader vehicles, while the increasing number of model parameters could in theory replicate better what is a complex phenomenon that combines components based strictly on the dynamics of the process with behavioral components. On the other hand, this makes it harder to find the right values of these parameters. "The calibration of the model" will be analyzed in detail in another section. Independent of the particular aspects of model calibration, there are at least three main questions that are intrinsically related to the models themselves:

– The stability of the car-following models
– The ability to replicate stop-and-go waves
– The ability to replicate capacity drops

The two key monographs on car following referenced in this chapter (Gerlough and Huber, 1975; Rothery, 1992) provide an overview on the stability analysis for linear car-following models. In summary, rescaling time in terms of the response time $t{=}\tau T$, eq. (1.19), can be rewritten as

$$\ddot{x}_{n+1}(\tau+1) = C\left[\dot{x}_n(\tau) - \dot{x}_{n+1}(\tau)\right] \tag{1.26}$$

where $C{=}\lambda T$. The solution of the differential equation (1.26) (Herman et al., 1959; Gazis et al., 1959) identifies the following conditions for the local behavior:

$0 \leq C \leq 1/e$ (0.368), spacing is nonoscillatory
$1/e \leq C \leq \pi/2$ (1.571), damped oscillation of spacing
$C = \pi/2$, spacing is oscillatory with undamped oscillation
$C > \pi/2$, increasing amplitude in oscillation in spacing

After a more detailed analysis of the values of C, the model (1.26) is generalized as

$$\ddot{x}_{n+1}(\tau+1) = C\frac{d^m}{dt^m}\left[x_n(\tau) - x_{n+1}(\tau)\right]$$

With $m = 0,1,2,3,\ldots$, Rothery (1992) concludes, "the results indicate that an acceleration response directly proportional to intervehicle spacing stimulus is unstable." These analyses have been extended for asymptotic stability, where platoons of vehicles are considered (Chandler et al., 1958), concluding that $\lambda T{<}0.5$ ensures stability, while $\lambda T{>}0.5$ would propagate a disturbance with increasing amplitude. These analyses have been extended to nonlinear car-following models by Del Castillo (1994).

A very detailed analysis of the stability of the Gipps' model was conducted by Wilson (2001) for the uniform flow solutions, concluding that uniform flow may become unstable only for unrealistic parameter values. Therefore the Gipps' model is not able to replicate stop-and-go waves. These results are similar to those of Abou-Rahme and White (1999), who simulated minor adaptations of the Gipps' model in an attempt to produce stop-and go waves, though they failed to find suitable parameter values.

Although a lot of improvements have been introduced in recent versions of traffic simulators based on these car-following models (e.g., decoupling the simulation step from reaction times, adding look-ahead abilities, making the estimation of the leader's deceleration more flexible), it still remains to be seen whether the reported drawbacks have been overtaken or not. Anyway, the main experimental results show that car-following models – and subsequently the simulation software based on them – provide reasonable results in uncongested conditions and in some cases in

congested conditions as well. But in accordance with the mentioned studies, they fail to provide results of a similar quality in the transitions from uncongested to congested, that is, when the steady-state hypothesis no longer holds. Maybe one of the reasons could be the lack of empirical evidence of enough quality for these conditions. This missing evidence has been partially filled by the NGSIM program of the Federal Highway Administration (NGSIM, 2002) and the new traffic data sets that have been made available to researchers and practitioners. In recent years a substantial amount of work has been devoted to modifying the existing models in order to improve the identified drawbacks or to refine the values of model parameters from the more accurate available traffic data. Examples of this could be the reports of Wang et al. (2005) or the *Special Issue on Traffic Flow Theory of the Transportation Research Board in 2007*. It is useful to have an overall view of the advantages and disadvantages in most of the currently available traffic simulation software from the perspective of car-following models. Panwai and Dia (2005) provide just such a comprehensive framework. Following Ranney (1999), they classify the factors that influence car-following behavior in the following categories:

- Individual differences: age, gender, risk-taking propensity, driving skills, vehicle size, vehicle performance characteristics
- Situational factors:
  - Environment: time of day, day of the week, weather, road conditions
  - Individual: Situations of distraction, hurrying, impairment due to alcohol, drugs, stress and fatigue, trip purpose, length of driving

And they propose the model displayed in Fig. 1.11 to represent the relative contribution of the different categories of factors that influence car-following behavior. According to this model, the car-following models described so far become active

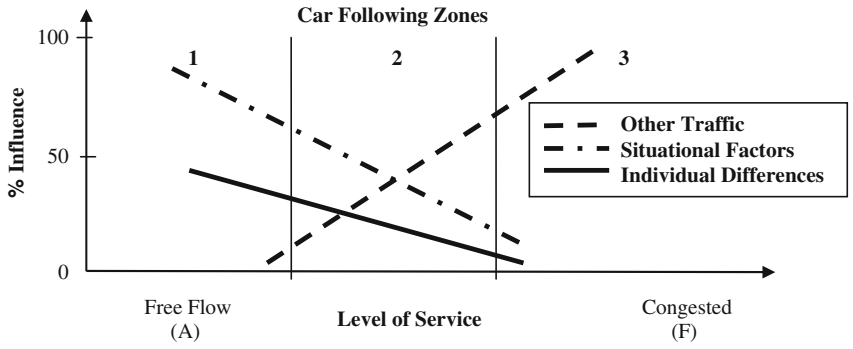


Fig. 1.11 Relative contribution of different factors in influencing car-following behavior (Panwai and Dia, 2005)

primarily at intermediate levels of service, represented by zone 2 in the figure. Under free-flow conditions, zone 1, car following occurs only if a driver chooses to closely follow other vehicle. Under congested conditions, driving is constrained and drivers have little choice about following the leader vehicle closely. But this model also shows that within zone 2, car following is determined by a combination of constraints imposed by other vehicles and individuals, as well as situational factors whose relative contribution could be a function of traffic congestion. It is under these conditions in the transition from 2 to 3 that car-following models perform poorly, and therefore it is where more research and development efforts should be concentrated.

The oversaturated freeway flow algorithm (Yeo and Skabardonis, 2007) can be considered as a significant contribution in this direction. After a detailed analysis of vehicle trajectories in congested no-passing freeway conditions (supplied by the NGSIM Traffic Data Repositories), they propose and test a model which is a combination of base car-following and lane-changing algorithms under various conditions. The model has been designed thinking of it explicitly as a component of a traffic simulation model. The model components are the following:

- Base car-following (CF) model;
- Merging car-following (MCF) model;
- Before lane changing (LC) car-following (BCF) model;
- During LC car-following (DCF) model;
- After LC car-following (ACF) model;
- Emergency LC car-following (ECF) model;
- Cooperation (COOP) car-following (CCF) model; and
- Receiving vehicle car-following (RCF) model.

The model structure is shown in Fig. 1.12.

A vehicle may be in any of the states shown in Fig. 1.12, but for the purposes of this section, we will restrict our focus only on the base car following. The base car following is derived from Newell's addition of safety constraints to avoid collisions, *when it is implemented in a simulation model and vehicle performance has limits on the rates of acceleration and deceleration*. The position for the follower vehicle n+1 for a time simulation step Δt is given as

$$x_{n+1}(t+\Delta t) = \text{Max}\left\{x_{n+1}^{\text{U}}(t+\Delta t), x_{n+1}^{\text{L}}(t+\Delta t)\right\} \tag{1.27}$$

where the upper bound is

$$x_{n+1}^{\text{U}}(t+\Delta t) = \text{Min}\{x_n(t+\Delta t-\tau_n) - l_n - g_n^{\text{jam}}, \quad \begin{array}{l}\text{acceleration capability,}\\ \text{maximum desired speed,}\\ \text{maximum safety distance}\}\end{array} \tag{1.28}$$

where the minimum value of four distances are as follows:

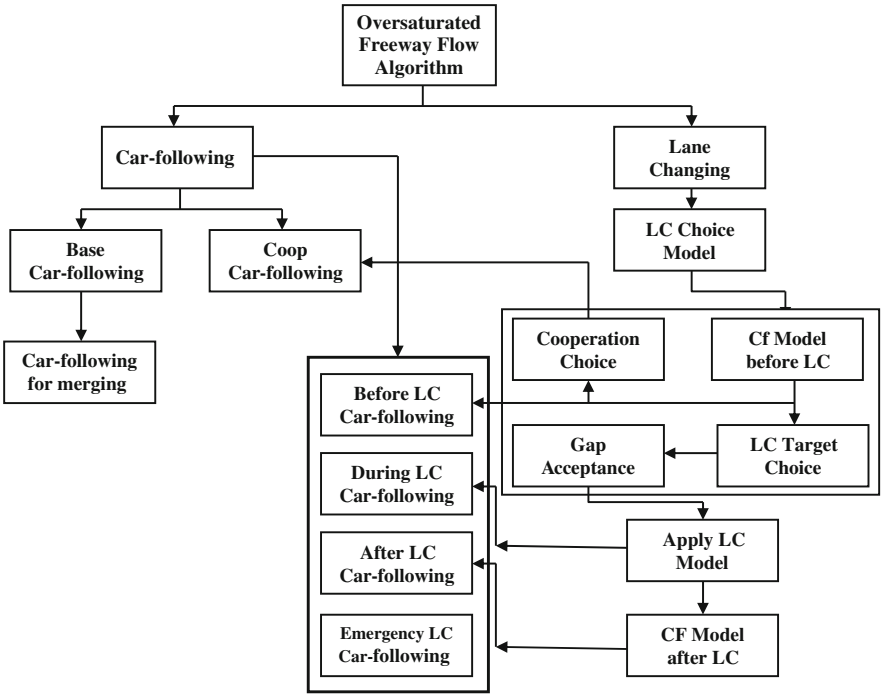


Fig. 1.12 Components of the proposed oversaturated freeway flow algorithm (Yeo and Skabardonis, 2007)

– Acceleration capability:

$$x_{n+1}(t) + \dot{x}_{n+1}(t)\,\Delta t + a^{\mathrm{U}}_{n+1}\Delta t^2 \tag{1.29}$$

– Maximum desired speed:

$$x_{n+1}(t) + v^{\mathrm{f}}_{n+1}\Delta t \tag{1.30}$$

– Maximum distance safety:

$$x_{n+1}(t) + \Delta x^{s}_{n+1}(t + \Delta t) \tag{1.31}$$

and the lower bound for the distance is determined by the deceleration capability and the current position of the vehicle. That is, the vehicle is not allowed to move backward:

$$x^{\mathrm{L}}_{n+1}(t + \Delta t) = \mathrm{Max}\left\{x_n(t) + \dot{x}_{n+1}(t)\,\Delta t + a^{\mathrm{L}}_{n+1}\Delta t^2, x_n(t)\right\} \tag{1.32}$$

where

$$x_{n+1}^{s}(t+\Delta t)=\Delta t\Bigg[a_{n+1}^{\mathrm{L}}\tau_{n+1}+\sqrt{\left(a_{n+1}^{\mathrm{L}}\tau_{n+1}\right)^{2}-2a_{n+1}^{\mathrm{L}}\left[x_{n}(t)-x_{n+1}(t)-\left(l_{n}+g_{n+1}^{\mathrm{jam}}\right)+d_{n}(t)\right]}\Bigg]$$

and

$$d_{n}(t)=-\frac{[\dot{x}_{n}(t)]^{2}}{2a_{n}^{\mathrm{L}}}$$

τ_n is the wave travel time for the vehicle n; l_n is the length of the vehicle n; g_{n+1}^{jam} is the jam gap between the nth vehicle and the n+1th vehicle, that is, the distance from the front end of a vehicle to the rear end of the preceding vehicle when both vehicles are stopped; $v_n^{\mathrm{f}}(t)$ is the free-flow speed of the nth vehicle at time t; a_n^{U} is the maximum acceleration of the nth vehicle; a_n^{L} is the maximum deceleration of the nth vehicle; and $\dot{x}_n(t)$ is its speed at time t. Additionally, the following equation must be satisfied:

$$x_{n+1}^{\mathrm{L}}(t+\Delta t)\leq x_{n+1}(t+\Delta t)\leq x_{n+1}^{\mathrm{U}}(t+\Delta t) \tag{1.33}$$

The realm of car-following models for non-steady-state conditions and non-constant accelerations is still a relatively unexplored territory. An analysis for linear accelerations was made by Aycin and Benekohal (1998); an extension of logarithmic acceleration models can be found in Mahut (1999); and Benekohal and Treiterer (1988) propose a specialization for stop-and-go conditions. But taking into account the conclusions of Wilson and others, who have already been mentioned, it is clear that this is still an open field for research. Wagner and Lubashevsky (2006) explore inconsistencies between empirical evidence on action points collected from equipped vehicles and the predicted behavior of car-following models; Lubashevsky et al. (2003) set some theoretical principles for an extension.

In practice, microscopic traffic simulation models are implemented as synchronous simulators that – at each simulation step Δt – explore all entities in the model and update the model's state by updating the entities' states. Concerning the definition of traffic demand as an input to the simulator, most of the existing simulators can operate in two alternative modes:

- The traffic demand input is defined in terms of input flows and turning proportions at intersection and exit sections in a similar way as that described for macroscopic models illustrated in Fig. 1.8. In this case there is neither routing in the model nor DTA. Vehicles travel stochastically in the network, leaving the network occasionally, according to the turning and exit proportions. This is the usual mode in most practical applications of microscopic simulation to small networks.
- The traffic demand input is defined in terms of origin–destination matrices as described in Section 1.2. Vehicles travel across the network from origins to

destinations along the available paths that join them. Route choice models and therefore DTA and/or DUE became an essential part of the model and the simulation study.

In terms of the computational framework described in Fig. 1.7, most of the current microscopic simulators fall into the category of those which iteratively perform the following generic process:

1 Initialization:
Define a cycle time to update paths in the network.
Calculate initial shortest paths for each OD pair on the basis of some definition of initial link costs, e.g., free-flow link travel times.
2 Repeat until all the demand has been loaded onto the network:

2.1 Calculate path flow rates according to a route choice model and the proportion of the demand for each OD pair for the selected cycle length.
2.2 Dynamic network loading: propagate the flows along the paths in accordance with the microscopic flow dynamics:

At every simulation step, update the position of every vehicle in the model:

- Determine the next move at the current simulation step and the associated model.
- Apply the corresponding model: lane change, car following, etc.
- Calculate the new position at the end of the simulation step.

2.3 Collect statistics according to a predefined data collection plan.
2.4 Update link costs.
2.5 Update shortest paths with the updated link costs, for use in the next cycle.

The main differences between the various microscopic simulators currently available lay in the way they implement this generic iterative process and combine or integrate their core car-following and lane-changing models. Figure 1.12 could be a good example of such integration. The details of the relevant cases included in this book will be described in subsequent chapters.

1.4.3 Mesoscopic Modeling of Traffic Flows

Mesoscopic modeling of traffic flow dynamics usually consists of a simplification that – while capturing the essentials of the dynamics – is less demanding of data and mesoscopic models are computationally more efficient than microscopic models. These approaches combine – in some ways – microscopic aspects (as far as they deal or can deal with individual vehicles) and macroscopic aspects, such as those concerning vehicle dynamics. Basically there are two main approaches to mesoscopic traffic simulation: those in which individual vehicles are not taken into account and vehicles are packed in packages or platoons (although platoons may

consist of one vehicle) that move along the links, as in CONTRAM (Leonard et al., 1989), and those in which flow dynamics are determined by simplified dynamics of individual vehicles, such as DYNASMART (Jayakrisham et al., 1994); DYNAMIT (Ben-Akiva et al., 1997, 2001, 2002); DTASQ (later on Dynameq) (Mahut, 2000; Florian et al., 2001, 2002; Mahut et al., 2003a, b, 2004); and MEZZO; (Burghout, 2004; Burghout et al., 2005). Another main difference between the mesoscopic approaches lays in the way they deal with time. The most common approaches are based on synchronous timing, that is, time-oriented simulations in which time in the model is advanced at an appropriately chosen unit time Δt, also known as the simulation step. This is the case for DYNASMART and Dynameq. Other approaches are asynchronous, or event based, that is, the state of the model changes when some events occur. Time is then advanced in variable amounts, depending on when such events occur. Dynameq and MEZZO are examples of event-based mesoscopic traffic simulators.

The existing approaches model the link, explicitly or implicitly, splitting it into two parts: the running part and the queue part (Fig. 1.13). The running part is that part of the link where vehicles are not yet delayed by the queue spillback at the downstream node, where the capacity is limited by stop, give way, or traffic lights.

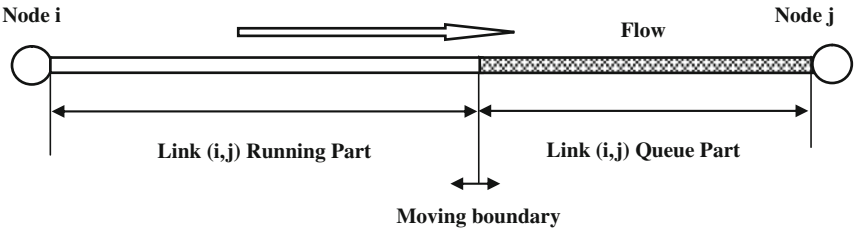


Fig. 1.13 Link model

Nodes are modeled according to the interactions between traffic flows at intersections, as node transfer modules, or to a queue server approach – in order to explicitly account for traffic lights and the delays that they cause (Mahmassani et al., 1994). Individual vehicle dynamics in the running part is approximated by a simplified car-following model that is compatible with the macroscopic speed–density relationship on the link. This speed is used to estimate the earliest time at which the vehicle could exit the link, unless it is affected by the queue spillback when reaching the border between the running part and the queue part. The vehicle dynamics is then ruled by the queue discharging process. The boundary between the running part and the queue part is dynamic, according to the queue spillback and queue discharging processes.

Various solutions have been proposed for simulating flow dynamics in the link running part. DYNASMART (Jayakrisham et al., 1994) determines link densities by solving the finite difference form of the continuity equation (1.6), given the densities and the inflows and outflows for each section at each time step. The section speeds are calculated from the densities using the modified Greenshield speed–density relationship:

$$u_i^t = (u_f - u_0)\left(1 - \frac{k_i^t}{k_{jam}}\right)^{\alpha} + u_0$$

where u_i^t and k_i^t are the mean speed and density in section i at time step t, respectively, u_f and u_0 are the mean free speed and the minimum speed, respectively, k_{jam} is the jam density, and α is a parameter that captures the sensitivity of speed to concentration. DYNAMIT uses the generalized speed–density relationship (1.8) of May and Keller (1967):

$$u = u_f\left[1 - \left(\frac{k}{k_{jam}}\right)^{\alpha}\right]^{\beta}$$

Other models, like MEZZO, complement this approach after empirical evidence establishes that there are two limiting densities k_{min} and k_{max}, which represent the minimum and maximum densities, respectively, where the speed is still a function of the density (Del Castillo and Benitez, 1995):

$$u = \begin{cases} u_f, & \text{if } k < k_{min} \\ u_0 + (u_f - u_0)\left[1 - \left(\dfrac{k - k_{min}}{k_{max} - k_{min}}\right)^{\alpha}\right]^{\beta}, & \text{if } k \in [k_{min}, k_{max}] \\ u_{min}, & \text{if } k > k_{max} \end{cases}$$

A completely different approach is taken in Dynameq based on a simulation model that moves vehicles individually, according to a simplified car-following model:

$$x_{n+1}(t) = \text{Min}\left[x_{n+1}(t - \varepsilon) + \varepsilon u_f, x_n(t - T) - L\right]$$

where $x_{n+1}(t)$ is the position of the follower vehicle n+1 at time t; T is the reaction time; u_f is the free-flow speed; L is the effective vehicle length; and ε is an arbitrary short time interval. The simplified model depends only on the free-flow speed. It does not consider accelerations and includes a simple collision avoidance rule. It can be shown (Mahut, 2000) that this model yields the triangular fundamental flow density model (Daganzo, 1994). The main events changing the state of the model are the arrivals of vehicles to links and the link departures – or transfers from one link to the next, according to the turning movements at intersections. For further details, the reader is referred to chapters 9 and 10 on Dynameq and DynaMIT, respectively, in this book.

The synthesized flow dynamics so far correspond to the dynamic network loading part described in the conceptual diagram in Fig. 1.7. To complete the mesoscopic simulator, the dynamic network loading mechanism must be combined with a route choice model. Various computational schemes have been proposed, as has been described in Section 1.3. For details, the reader is referred to how they are implemented in the mesoscopic models in this book (see chapters 9 and 10). However, to complete the description, we will briefly describe one of the most typical algorithmic approaches: the method of successive averages (MSAs). The MSA procedure

redistributes the flows among the available paths in an iterative procedure that, at iteration n, computes a new shortest path $sp_{rs}(t)$ from origin r to destination s at time interval t. Then, if $P_{rs}^{n}(t)$ is the set of paths from origin r to destination t at time interval t and iteration n, $p \in P_{rs}^{n}(t)$ is a path from r to s, and $d_{rs}(t)$ is the demand from r to s at time interval t; the path flow update process is as follows:

Case a $sp_{rs}(t) \notin P_{rs}^{n}(t)$

$$f_{rsp}^{n+1}(t) = \begin{cases} \alpha_n f_{rsp}^{n}(t), & \text{if } p \in P_{rs}^{n}(t) \text{ and } p \neq sp_{rs}(t) \\ \forall r, s, t & \\ (1-\alpha_n)\, d_{rs}(t), & \text{if } p = sp_{rs}(t) \end{cases}$$

Let $P_{rs}^{n+1}(t) = P_{rs}^{n}(t) \cup sp_{rs}(t)$

Case b $sp_{rs}(t) \in P_{rs}^{n}(t)$

$$f_{rsp}^{n+1}(t) = \begin{cases} \alpha_n f_{rsp}^{n}(t), & \text{if } p \neq sp_{rs}(t) \\ & \forall r, s, t \\ \alpha_n f_{rsp}^{n+1}(t) + (1-\alpha_n)\, d_{rs}(t), & \text{if } p = sp_{rs}(t) \end{cases}$$

Let $P_{rs}^{n+1}(t) = P_{rs}^{n}(t)$

Depending on the values of the weighting coefficients α_n, different MSA schemes can be implemented (Carey and Ge, 2007). Perhaps the most typical value is $\alpha_n = n/(n+1)$. Varia and Dhingra (2004) propose an interesting modified MSA algorithm, where the weighting coefficient takes into account a variable step length which depends on the current path travel times:

$$\alpha_n = \frac{\lambda k \left[\exp\left(-\tau_{rsp}(t)\right)\right]}{(n+1)\left[\sum_{p}\left[\exp\left(-\tau_{rsp}(t)\right)\right]\right]}$$

One of the potential computational drawbacks of these implementations of MSA is the growing number of paths in the case of large networks. To avoid this, several modified implementations have been proposed (Peeta and Mahmassani, 1995; Sbayti et al., 2007). However, possibly one of the most computationally efficient implementations is the one proposed by Florian et al. (2002), which keeps the number of alternative paths bounded in order to account for each origin–destination pair. If K is the maximum number of paths to keep, then the algorithm proceeds as before, by alternating cases a and b until the K paths are reached. It then proceeds only as in case b. This variant of the algorithm initializes the process on the basis of an incremental loading scheme, distributing the demand among the available shortest paths. The process is repeated for a predetermined number of iterations after which no new paths are added and the corresponding fraction of the demand is redistributed according to the MSA scheme. However, taking into account the possibility of repeating shortest paths from one iteration to the next – in order to

keep a maximum of K different shortest paths – a proper implementation of the algorithm requires that the number of iterations n is defined by O–D pair and time interval.

All the proposed approaches for DUE are based on simulation procedures for the network loading process and therefore are heuristic in nature. Therefore no formal proof of convergence can be provided. Consequently, there is a way of empirically determining whether the solution reached can be interpreted in terms of a DUE, in the sense that "the actual travel time experienced by travelers departing at the same time are equal and minimal." This can be based on an ad hoc version of the relative gap function proposed by Janson (1991):

$$\text{Rgap}(n) = \frac{\sum_{t} \sum_{(r,s)\in\Im} \sum_{p\in P_{rs}(t)} f^{n}_{rsp}(t) \left[\tau^{n}_{rsp} - \theta^{n}_{rs}(t)\right]}{\sum_{t} \sum_{(r,s)\in\Im} d_{rs}(t)\, \theta^{n}_{rs}(t)} \tag{1.34}$$

where $f^{n}_{rsp}(t)$ is the flow on path p from r to s departing origin r at time t at iteration n, and the difference $\tau^{n}_{rsp}(t) - \theta^{n}_{rs}(t)$ measures the excess cost experienced by the fact of using a path of cost $\tau^{n}_{rsp}(t)$ instead of the shortest path of cost $\theta^{n}_{rs}(t)$ at iteration n. The ratio measures the total excess cost with respect to the total minimum cost if all travelers were to have used shortest paths.

1.5 Calibration and Validation of Traffic Simulation Models

Simulation is a technique that can be seen as a *sampling experiment* on a dynamic real system through a computer model formally representing it. Simulation assumes that the evolution of the system's model over time properly imitates the evolution of the modeled system over time. Thus, samples of the observational variables of interest are collected. From these samples, conclusions on system behavior can be drawn by using statistical analysis techniques. In order to use the model as an experimental substitute for the actual system, the reliability of this decision-making process depends on the ability to produce a simulation model that represents the system's behavior closely enough (Barceló and Casas, 2004; Dowling et al., 2004; FHWA, 2004). The process of determining whether the simulation model is close enough to the actual system is usually achieved through the validation of the model and the insight gained, in order to improve the model until the accuracy is judged to be acceptable. Validation of the model is an iterative process that calibrates the model parameters, compares the model to the actual system behavior, and uses the discrepancies between the two and the insight gained, to improve the model until the accuracy is judged to be acceptable. Validation is therefore concerned with determining whether the simulation model is an *accurate representation of the system under study.* The calibration process has the objective of finding the values of these parameters that will produce a valid model. Model parameters must be supplied with values. *Calibration is the process of obtaining such values from field data in a particular setting*.

The question of whether a model is valid or not can be formulated in terms of whether model results faithfully represent reality, a question for which statistical techniques provide a quantified answer. Quantification, according to Rouphail and Sacks (2003), can be formally stated in the following terms: the probability that the difference between the "reality" and the simulated output is less than a specified tolerable difference within a given level of significance:

$$P\{\ |\text{“reality”} - \text{simulated output}|\ \leq d\ \} > \alpha$$

where d is the tolerable difference threshold indicating how close the model is to reality, and α is the level of significance that indicates the certainty of the result. In this framework, the analyst's perception of reality relies on the information gathered through data collection and subsequent data processing in order to account for uncertainties. *The available data and its uncertainties will determine what can be said about d and* α. Surprisingly the last assertion has received little attention. In almost all the discussions and methodological approaches for calibration and validation of traffic simulation models, attention has been focused on the processes' ability to accurately estimate model parameters and on the statistical methods for assessing model validity; meanwhile, the implicit assumption has been that the available data for comparison were reliable enough. Figure 1.14 provides a deeper insight into the

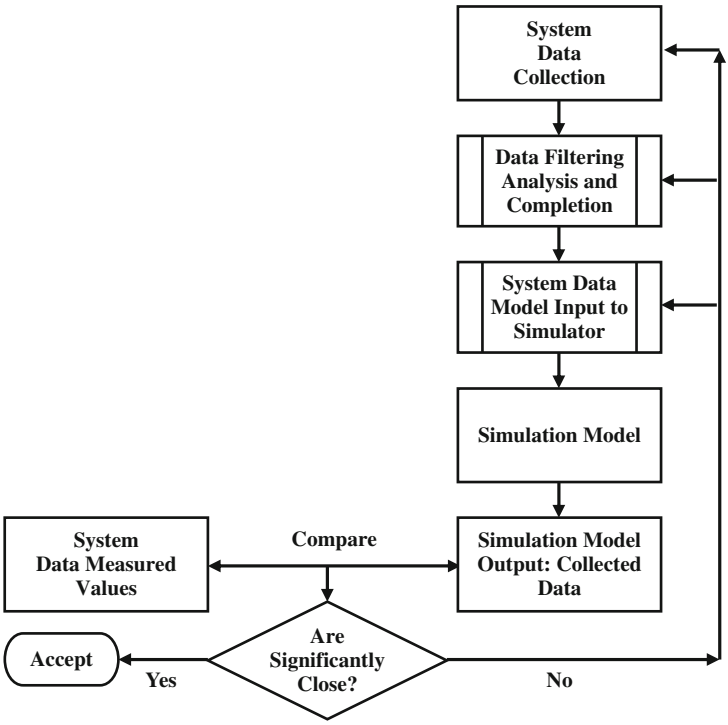


Fig. 1.14 Methodological scheme for validation of simulation models

validation steps of Fig. 1.1. It conceptually synthesizes the basic methodological approaches for the validation of simulation models. This methodology implicitly assumes that

- we are able of properly model the input data and
- the set of measured data to compare with the simulation results are reliable or, in other words, we are assuming that they are error free.

Data inputs to traffic models can be classified in two categories:

- Directly observable data, i.e., measurements of traffic variables affected by errors (flows, speeds, occupancies, travel times, etc.), which are based on available technologies and which must be suitably filtered and processed before using them in the applications (Bayarri et al., 2004).
- Data not directly observable, such as transport demand modeled in terms of time sliced into origin–destination matrices. This input process asks for sound, indirect estimation procedures in order to generate the suitable inputs.

The methodological approach also assumes that after suitably processing the system's collected data, filtering out the unreliable measurements, and completing the missing data, a set of data are available either for direct input to the simulation models (e.g., when the data input to the simulation model is based on input flows and turning proportions as mentioned earlier) or for building an appropriate input data model for the simulation model (e.g., when the simulation data input consists of an origin–destination matrix). In essence, the validation process consists of collecting the simulated data and comparing it to the system's measured data, based on statistical analysis methods, to determine whether the samples of observed and simulated data are significantly close enough, as stated above. In the case of a positive answer, the simulation model is accepted as valid; otherwise it is rejected and the validation process has to be revised. This implies an iterative procedure in which, depending on the situation, more or perhaps new data have to be collected; the data processing has to be revisited; or the model of the input data has to be changed or refined.

Calibration and validation of simulation models is still a major challenge in the use of simulation for practical purposes, namely in the case of microscopic traffic simulation models that combine the high level of uncertainty of the modeled system with a large number of parameters. Some of them account for behavioral aspects of the vehicle–driver system. Consequently, calibration and validation has attracted the attention of many researchers in recent years and the increasing use of microscopic traffic simulation for traffic analysis has prompted some initiatives from researchers (Yoshii, 1999; Bayarri et al., 2002; Dowling et al., 2004; Ciuffo et al., 2007; Hollander and Ronghui, 2008) and governmental agencies (FHWA, 2004) to develop methodological guidelines for calibration and validation of microscopic simulation models. In general all methodological guidelines coincide in recommending the decomposition of the main calibration problem into sub-problems to

solve it more efficiently, taking into account the different nature of the parameters to be calibrated. Some are local, while others are global. However, as it will be discussed later on in this section, the availability of more powerful computational techniques makes the resort to simultaneous procedures more appealing (Hollander and Liu, 2008). Nevertheless, in most cases it is still highly recommended to decompose the problem into simpler sub-problems prior to applying a simultaneous procedure. This practice is usually of great help in determining the likely intervals of parameter variability and constitutes a useful input for simultaneous procedures. In accordance with these recommendations, the FHWA (2004) guidelines structure the process in four stages:

1. *Error checking*: The coded transportation network and demand data are reviewed for errors. This step is necessary for weeding out coding errors before proceeding with calibration.
2. *Capacity calibration*: An initial calibration is performed to identify the values for the capacity adjustment parameters that cause the model to best reproduce observed traffic capacities in the field. A global calibration is first performed, followed by link-specific fine-tuning. The highway capacity manual can be used as an alternative source of capacity target values if field measurements are infeasible.
3. Route choice calibration: If the microsimulation model includes parallel streets, then route choice will be important. In this case, a second calibration process is performed, but this time with the route choice parameters. A global calibration is first performed, followed by link-specific fine-tuning.
4. Performance validation: Finally, the overall model estimates of system performance (travel times and queues) are compared to field measurements of travel times and queues. Fine-tuning adjustments are made to enable the model to better match the field measurements.

This framework is not necessarily sequential; once step 1 has been satisfactorily achieved, steps 2–4 can be iteratively repeated in case there is no acceptable match of the comparison criteria between the selected measures of performance and the field data, i.e., volumes and speeds, location of congestions, queue discharges, delays, and others.

Zhang and Ma (2008) propose a more detailed procedure in which the capacity calibration step is split into a "global parameter and a local parameter calibration." The first is aimed at adjusting the default driving behavior parameters for typical road sections and the second specializes in fine-tuning site-specific driving behavior parameters at critical locations. They also refine the "route choice" calibration step, proposing first a calibration of the route choice and departure time behavior with a fixed demand and then proceeding to a refinement of the demand. The report discusses a variety of algorithmic procedures; some of them will be analyzed later on.

Error checking: Manual error checking for large networks is a cumbersome and error-prone task. Most of the available commercial traffic simulation software

provide advanced user-friendly graphic user interfaces with flexible and powerful graphic editors to assist analysts in the model-building process. This reduces the number of errors but in no way does it eliminate them. What is worse, the remaining network coding errors are the most difficult to find visually or manually. At the same time, those are the ones which have more influence in the model response, i.e., the network connectivity which ensures that there are paths from every origin to every destination, parameters governing the dynamics of vehicles at turnings or the queue discharging processes at intersections, and so on. Some of them (see chapters 2–8 in this book) also include tools designed to analyze the topology of the network representation in order to automatically identify network coding errors (for example, data missing in a section, number of lanes, and capacity). These tools are also used for detecting errors in the connectivity of the networks, either in terms of connections between the sections and their lanes or with respect to origin and destination nodes, i.e., the existence of feasible paths between all centroids representing origin and destination zones in the network. Some other errors can be detected only when running the simulation. Such errors are model inconsistencies which make the vehicles behave improperly in a blatant fashion that leads to grid locks – for example, in the case of a signalized intersection with an improper definition of the control plan; situations in which parameters which govern queue discharges are incorrectly set and lead to capacity reductions at intersection turnings; and inappropriate specifications of give-way or lane-changing parameters, which stop the vehicle at certain positions for unacceptably long periods.

Capacity calibration: Quite frequently, this step in the process consists of determining the most appropriate values of the main parameters of the core models, car following and lane changing, to accurately reproduce point observations of traffic variables, flows, speeds, and densities. This is a calibration methodology based on the ability of car-following models to reproduce macroscopic flow variables already discussed in Section 1.4.1. The exploitation of this property-to-ground model calibration is quite natural and takes into account that aggregated point measurements are usually the only ones available in real-life projects. The detailed analysis of trajectories is key for a deeper insight into vehicle behavior and a better understanding of how the models work, which enables the development of new models, as in the case of the NGSIM oversaturated freeway flow model (Yeo and Skabardonis, 2007). Despite this, these types of data are still of no use in real-life projects, as they are expensive and time consuming to obtain. Therefore in real life, calibration has to be based on aggregated macroscopic data; the most reliable methodological process consists of checking the values of the model parameters that are better fitted to the fundamental relationships of speed–density, flow–density, speed–flow, speed at capacity, capacity, jam density, and, in some cases, speed–headway. Rakha et al. (2007) conduct a detailed analysis of the dependencies of the Gipps' model in a stationary state with the critical parameters of traffic stream free-flow speed, the most severe deceleration rate b of the follower vehicle in order to avoid collision and the follower's estimate of the most severe breaking of the leader $\hat{b}$. After performing a theoretical analysis based on Wilson's (2001) work, they found that for the case when $b = \hat{b}$, the space headway acts as the stimulus and the sensitivity factor can

be computed as a function of the fundamental traffic variables q_c, the road capacity expressed in terms of vehicles per hour per lane, k_{jam}, the jam density measured in terms of vehicles per kilometer per lane, and u_f, the traffic stream free-flow speed. From this, the reaction time T in seconds can be computed as follows:

$$T = 2,400\left(\frac{1}{q_c} - \frac{1}{k_{jam}u_f}\right)$$

This formula can be used to calibrate driver reaction time as a function of the macroscopic traffic stream parameters or, in other words, the reaction time that has to be used in the Gipps' car-following model for reproducing the values of q_c, k_{jam}, and u_f when $b = \hat{b}$.

When $b > \hat{b}$, this case may yield speed–headway that becomes unphysical, according to Wilson. Rakha et al. prove that in this case the user must choose parameters that satisfy the relationship

$$u_f < 2.4\frac{Tb}{\left(\frac{b}{\hat{b}} - 1\right)}$$

This holds for ratios of b to $\hat{b}$ very close to 1, but then the behavior of the model becomes unrealistic and therefore is not recommended for professional use. The study concludes by analyzing the case when $b < \hat{b}$. This could be interpreted as the case when the follower overestimates the leader's deceleration. This case leads to the following general formulation of the speed–headway relationship that combines congested and uncongested regimes:

$$u = \min\left(u_f, \frac{5.4bT}{\left(1 - \frac{b}{\hat{b}}\right)}\left[1 + \sqrt{1 + \frac{8\left(h - h_{jam}\right)\left(1 - \frac{b}{\hat{b}}\right)}{9bT^2}}\right]\right)$$

where h_{jam}, the minimum space headway between two vehicles, is the inverse of the jam density. From this relationship the other relationships for speed–density or capacity can be derived (see Rakha et al., 2007 for details), as well as for tables with various combinations of values for u_f, k_{jam}, and the ratio $b/\hat{b}$. These combinations can be used by practitioners to select the combination most suited to their observations. Rakha and Gao (2009) extend these results to other car-following models.

The calibration procedures for car-following parameters discussed so far, although interesting and useful for practitioners, have some drawbacks when applied to situations other than the steady-state traffic in the main section of a freeway, e.g., at freeway merges. Stewart (2003) tried to analyze the time needed to break down an important factor by determining capacity loss after the flow breakdown, which dictates the performance of the freeway. The simulation results showed the difficulty in properly calibrating the model. One of the difficulties lay in finding the right

reaction time that –if calibrated for the steady-state conditions – overestimated the capacity at breakdown while – if calibrated at the merge zone—underestimated the capacity at the other basic freeway segments. Another difficulty was that it became clear whether other parameters had to be taken into account, like the length of the acceleration lane and the parameters governing the merging maneuvers. Hidas (2002, 2004) conducted extensive simulation studies to analyze the performance of simulation models in various situations. For example, queue discharge when the car-following characteristics (e.g., mean headways) are significantly different from the average characteristics; when lane changes under congested flow conditions may lead to significantly shorter vehicle spacing; how to account for vehicle interactions at on-ramp merging; or for modeling the change in driving behavior in emergencies. Similar results have been reported more recently by Liu and Hyman (2008), who show that the gap acceptance approaches used in many microsimulation models underestimate the capacity at merges and, therefore, overestimate the associated delays. Simulation software developers have worked significantly on introducing model changes in order to address some of these issues (see forthcoming chapters for details on specific developments). Lane change has been the object of specific research in NGSIM (Choudhury, 2007; Choudhury et al., 2007, 2008), but it is still unclear whether or not most of the available simulators can address the above questions in a standard way.

Anyway what is clear is that, although necessary and recommendable, the calibration exercise described so far is not enough. Taking into account the usually large number of parameters on which the microscopic models depend, the methodological recommendation is that the analyst should select an objective function in terms of one of the selected measures of performance and the model parameters on which it depends, but only after dealing individually with each of them in a preliminary stage in order to determine the most likely bounds of the intervals of potential variation of parameter values.

In the case of the Gipps' car-following model, Punzo and Tripodi (2007) extended it for the multiclass, multilane case after conducting a similar analysis to that of Rakha et al. (2007) of the Gipps' model. Their analysis, for the case of a multilane freeway and a traffic mix of heavy and light vehicles, with a proportion of α heavy vehicles yields the lane speed as a function of the percentage of heavy vehicles and the Gipps' model parameters for each class:

$$v = \min\left\{V, f\left(q, \alpha, \boldsymbol{\beta_1}, \boldsymbol{\beta_2}\right)\right\}$$

where $\boldsymbol{\beta_1}$ and $\boldsymbol{\beta_2}$ are the vectors of parameters for each class of vehicles, $\beta_j = \left(T_j, \theta_j, b_j, \hat{b}_j, h_{\text{jam}}\right)$, $j = 1, 2$ where, as defined in Section 1.4.2, T_j is the reaction time of class j, θ_j the safety margin, b_j and $\hat{b}_j$ are the most severe deceleration rate of the follower vehicle in order to avoid collision and the follower's estimate of the most severe breaking of the leader, respectively, and h_{jam} is the minimum space headway between two vehicles of class j. For the explicit form of function $f(q, \alpha, \boldsymbol{\beta_1}, \boldsymbol{\beta_2})$, see Punzo and Tripodi (2007). Then, the calibration procedure is defined in

terms of the optimization problem:

$$\min_{\boldsymbol{\beta}} \left\{ \sum_i (v_i - \tilde{v}_i)^2 \right\}$$

where $\boldsymbol{\beta}$ is the vector of parameters to be calibrated, $\boldsymbol{\beta}=(\boldsymbol{\beta_1}, \boldsymbol{\beta_2}, V_1, V_2)$, V_1 and V_2 are the target speeds of classes 1 and 2 of vehicles, v_i is the model lane speed, and $\tilde{v}_i$ is the observed speed for the ith time interval. To account for the speed differences between classes, $\tilde{v}_i$ is estimated according to Daganzo (1997):

$$\tilde{v}_i = \frac{1}{\frac{\delta}{\tilde{v}_i^1} + \frac{(1-\delta)}{\tilde{v}_i^2}}, \text{where} \delta = \frac{\tilde{q}_i^2}{\tilde{q}_i^{tot}}$$

$\tilde{v}_i^1$ and $\tilde{v}_i^2$ are the observed speeds for vehicles of classes 1 and 2, respectively, $\tilde{q}_i^2$ is the observed flow for class 2, and $\tilde{q}_i^{\text{tot}}$ is the total observed flow at the ith time interval.

The formulation of the model's calibration process as an optimization problem is perhaps the most recommended practice. The optimization problem minimizes an objective function, expressing the "distance" between an observable traffic variable and its simulated value, constrained by the set of feasible values of the model parameters on which the simulated variable depends; Hourdakis et al. (2003) and Ma et al. (2007) could be examples of that. Hollander and Liu (2008) provide a comprehensive review. In the case of Hourdakis et al. (2003), the optimization problem is formulated as

$$\begin{aligned} &\text{Min} F = \sum_{j=1}^{\text{st}} \sum_{i=1}^{m} \left(v_{si}^j - v_{oi}^j \right)^2 \\ &\text{subject to} : L_{xp} \le x_p \le U_{xp}, \quad p = 1, 2, ..., n \end{aligned}$$

where st is the number of counting stations, m is the number of time intervals, n is the number of simulator parameters to be optimized, L_{xp} and U_{xp} are the lower and upper bounds of parameter x_p, respectively. The function to be optimized is the square of the differences between the simulated value v_{si}^j and the observed value v_{oi}^j at counting station j and time interval i. A variant of this optimization method proposed in FHWA (2004) is the following: let q_{ltr} be the simulated flow at location l and time t for replication r of the simulation and let MSE, the mean squared error, be the sum of the squared error averaged over the number of independent replications R of the simulation, e.g., replications using different random seeds. Let p be the set of model parameters whose values are the object of the calibration exercise. If q_l are the measured flows at location l and the time t, the determination of the most suitable set of parameter values can be formulated in terms of the following optimization problem:

$$\text{Min}\left\{\text{MSE} = \frac{1}{R}\sum_{r}(q_{ltr} - q_l)^2\right\}$$
$$\text{subject to} p_m^{\min} \leq p_m \leq p_m^{\max}$$

where p_m is the value of parameter m and $p_n^{\min}$ and $p_n^{\max}$ are the upper and lower bounds of the allowable range of values, respectively, for parameter p_m estimated at the preliminary analysis. This variant introduces a new element into the calibration practice: the optimization not only over a number of time intervals, to account for the time variability of traffic flows, but also over a certain number of independent replications of the simulation. This proposal accounts for a well-known fact: the strong dependency of simulation on the variability induced by the random seeds used to generate the samples. Smoothing out the variability by averaging over a number of independent replications is a sound practice. In essence, the optimization approaches to model parameter calibration can be generalized as follows:

$$\begin{aligned} &\min_{\boldsymbol{\beta}} f\left(M_{\text{obs}}, M_{\text{sim}}\right) \\ &\text{subject to}: L_i \leq \beta_i \leq U_i, \quad i = 1, 2, ..., n \end{aligned} \tag{1.35}$$

where f is a function of the observed and simulated measures of performance, M_{obs} and M_{sim}, whose arguments are the components of vector $\boldsymbol{\beta}$, each one is bound from below and above by upper and lower bounds L_i and U_i, respectively. This optimization framework is likely the most widely accepted, but it raises a set of key questions whose answers have provided most of the variants that are currently used:

- Which is the most adequate objective function for measuring the degree of closeness between simulated and measured values?
- Should the calibration rely on individual point measurements for each location or is it more adequate to propose global indices which provide an overall insight?
- How can the upper and lower bounds of parameter values be estimated?
- Which is the most appropriate optimization method for determining the most suitable parameter values?
- Is it appropriate to calibrate capacity and route choice independently?

Hollander and Liu (2008) provide a complete critical review of the measures of goodness of fit used by the different calibration methodologies as objective functions. They also conclude that those measures that depend on the squared differences penalize large errors, which are more appropriate given that penalizing small error would be wrong and lead to over-specified models, considering that small fluctuations over the mean are in the nature of traffic phenomena. Among the most used measures, if x_i and y_i are the ith measured and observed value, respectively, one can highlight the following:

The root mean square error, which quantifies the overall error:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2}$$

The root mean squared normalized error, which provides information on the magnitude of the errors relative to the average measurement:

$$\text{RMSNE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - y_i}{y_i}\right)^2}$$

Two other used measures of goodness of fit (Toledo and Koutsopoulos, 2004) are

The mean error:

$$\text{ME} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)$$

and the mean normalized error:

$$\text{MNE} = \frac{1}{N}\sum_{i=1}^{N}\frac{x_i - y_i}{y_i}$$

which are useful when applied separately to measurements at each location instead of to all measurements jointly. They indicate the existence of systematic bias in terms of under- or overprediction by the simulation model.

Despite recognizing the significance of individual measurements, many analysts consider it more useful to use joint measures that provide an overall view. One that is widely accepted by practitioners is the Geoffrey E. Havers' statistic GEH (Highways Agency, 1996), which calculates the index for each counting station:

$$\text{GEH}_i = \sqrt{\frac{2\,(x_i - y_i)^2}{x_i + y_i}}$$

It then estimates an aggregated index by means of the following algorithm:

For $i = m$ (number of counting stations)
    If $\text{GEH}_i \leq 5$, then set $\text{GEH}_i = 1$
        Otherwise set $\text{GEH}_i = 0$
    Endif;
End for;

Let GEH $= \frac{1}{m} \sum_{i=1}^{m} \text{GEH}_i$
If GEH $\geq 85\%$ then accept the model
Otherwise reject the model
Endif;

which can be interpreted in the following terms: if the deviation of the simulated values with respect to the measurement is smaller than 5% in at least 85% of the cases, then accept the model. There is neither theory behind the method nor theoretical criteria to determine the value thresholds that are purely empirical based on practice.

Taking into account that the series of measurements and simulated values can be collected at regular time intervals, it becomes obvious that they can be interpreted as time series and, therefore, used to determine how close the simulated and the observed values are. It is equivalent to determining how similar both time series are. On the other hand, the use of aggregated values to validate a simulation seems contradictory if one takes into account that it is dynamic in nature, and thus time dependent. Consequently, other analysts propose statistical methods which account specifically for the comparison of the disaggregated time series of observed and simulated values. Theil (1961) defined a set of indices aimed at this goal and these indices have been widely used for that purpose (Hourdakis et al., 2003; Barceló and Casas, 2004; Toledo and Koutsopoulos, 2004; Hollander and Liu, 2008). The first index is Theil's indicator (also called Theil's inequality coefficient), which provides a normalized measure of the relative error that smoothes out the impact of large errors:

$$U = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i)^2} + \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i)^2}}$$

The global index U is bounded, $0 \leq U \leq 1$, with $U = 0$ for a perfect fit and $x_i = y_i$ for $i = 1$–N, between observed and simulated values. For $U \leq 0.2$, the simulated series can be accepted as replicating the observed series acceptably well. The closer the values are to 0, the better. For values greater than 0.2, the simulated series is rejected.

Theil's indicator can be decomposed into three proportions:

$$U_{\text{m}} = \frac{N (\bar{x} - \bar{y})^2}{\sum_{i=1}^{N} (x_i - y_i)^2}, \; U_{\text{s}} = \frac{N \left(\sigma_x - \sigma_y\right)^2}{\sum_{i=1}^{N} (x_i - y_i)^2}, \; U_{\text{c}} = \frac{2N (1 - \rho) \sigma_x \sigma_y}{\sum_{i=1}^{N} (x_i - y_i)^2}$$

where $\bar{x}$ and $\bar{y}$ are the means of observed and simulated values, respectively, σ_x and σ_y are their standard deviations, and ρ is the correlation coefficient. They satisfy the relationship $U_m + U_s + U_c = 1$. Theil's bias proportion U_m is a measure of systematic error; high values close to 1 reveal an unacceptable bias. U_s is Theil's variance proportion and it measures how well the simulated values replicate the variability of the observed series. High values of U_s close to 1 indicate that the simulated series has a significantly different variability. Finally U_c, the covariance proportion, measures the remaining unsystematic error and consequently it should be close to 1 for a good fit.

The other pending question concerns the estimation of the ranges of variability of the parameter values in the calibration process, that is, upper and lower bounds that have to be used in the optimization models. The analytical procedures discussed above for the case of car-following parameters provide an answer to this question when the analysis is constrained to a limited set of basic parameters. However, for more complex situations, when much more parameters are taken into account, analytical methods cannot be used as long as there are no analytical relationships among them. A sound but computationally costly methodological approach consists of using statistical techniques from the factorial analysis design of experiments (Law and Kelton, 1991). Ciuffo et al. (2007) identify the relevant factors, that is, the parameter object of the calibration process, and they propose candidate ranges of variation for the parameter values and also their discrete increments, i.e., 0.1-s increment for the reaction time. They make pilot runs independently replicating the simulations for each combination of values and conduct an ANOVA variance analysis to complete the study. This procedure also allows identification of the degree of significance of the parameters and what could be even more relevant in certain cases of the combined effects of the parameters. Barceló and Casas (2006) and Ciuffo et al. (2007) present some practical examples for the application of this technique.

Route choice calibration: All methods described so far are explicitly or implicitly assuming that the model object of study is a relatively simple one, more or less linear, as it is in the case of freeways, where no routing exists. Therefore, the calibration exercise addresses car-following and lane-changing parameters as the only relevant parameters. However, when models are more complex, as is usually the case for road networks, namely urban networks, they have to account for vehicles traveling across the network from origins to destinations along paths connecting them (as has been discussed in Section 1.2). In this situation, dynamic traffic assignment becomes a key component of the model's ability to explain how flows are distributed and progress along the network. The traffic flow model becomes the core component of the network loading and the simulator is completed with a path calculation module and its complement, the path flow rate calculation (as described in Section 1.3). Most traffic simulators implement route choice models by explicitly or implicitly assuming a reactive approach (as discussed in Section 1.4.2), in which route choice models are usually based on discrete choice theory that emulates human behavior in choosing an alternative. The chosen alternative is the path along which the vehicle will travel. The perceived utility in the route choice model is usually the experienced travel time, although many simulators accept generalized travel costs to measure the

utility. The most used models are logit, modified logit models, and something similar, and they depend on behavioral parameters that have to be calibrated. This is usually a difficult task, quite frequently done by trial and error as factorial design approaches become more complex through the increasing number of factors and combinations. Barceló and Casas (2006) discuss some practical cases.

The problem is that, in practice, improperly calibrated route choice models yield inappropriate flow distributions on the road network. Also, they quite often produce undesirable, sudden flip-flop route changes that do not correspond to realistic behaviors. Fox (2008) critically analyzes the current built-in route choice models in many of the currently available microsimulators, such as ad hoc dynamic assignment models that are somewhat experimental and not based on sound equilibrium principles. His critics conclude that "micro-simulation is inappropriate for any networks where route choice is a key issue."

These route choice issues do not appear when the path flow rates are computed on the basis of a dynamic equilibrium paradigm, as in the case of the mesoscopic models (see Mahut et al., 2004, 2008, for instance). This has prompted an investigative interest in whether or not the iterative schemes of DUE can also be implemented when the network loading is based on a microscopic simulation. For details, the reader is referred to the DTA sections of chapters 5 to 11 in this book. Barceló and Casas (2006) and Liu et al. (2005) explore this issue by implementing an iterative scheme in which link costs are estimated as

$$c_{it}^{k+1} = \lambda c_{it}^{k} + (1 - \lambda)\, \tilde{c}_{it}^{k} \tag{1.36}$$

where c_{it}^{k+1} is the cost of using link i at time t at iteration k+1, and c_{it}^{k} and $\tilde{c}_{it}^{k}$ correspond to the expected and experienced link costs, respectively, at this time interval from previous iterations. This sets up a computational framework for a heuristic algorithm to compute dynamic equilibrium. Convergence to a state that can be interpreted in terms of equilibrium is verified empirically by the Rgap function (1.34) (Janson, 1991). However, there is no guarantee of convergence for this combination of stochastic route choice and iterative process unless route choice parameters and smoothing parameter λ are carefully calibrated. The set of results displayed graphically in Fig. 1.15 shows that in these cases, one performance criterion is not enough. In the computational experiments GEH and Rgap have been used.

The figure plots GEH versus Rgap for a set of experiments in which a C-logit route choice function has been combined in the iterative scheme with the link cost function (1.36). The results show that if only GEH is used, results can be achieved in which the model could be considered "calibrated" in terms of its ability to reproduce the observed flow at a number of counting stations, in accordance with, the GEH criterion discussed above. From the point of view of the Rgap, flows are far from a dynamic equilibrium, as the big value of Rgap indicates that path costs for some origin–destination pairs can be quite unbalanced. Conversely, situations can be achieved in which path costs are well balanced and the small value of Rgap could potentially be interpreted in terms close to an equilibrium, while GEH is telling us that the simulated values for an unacceptable number of counting stations are not

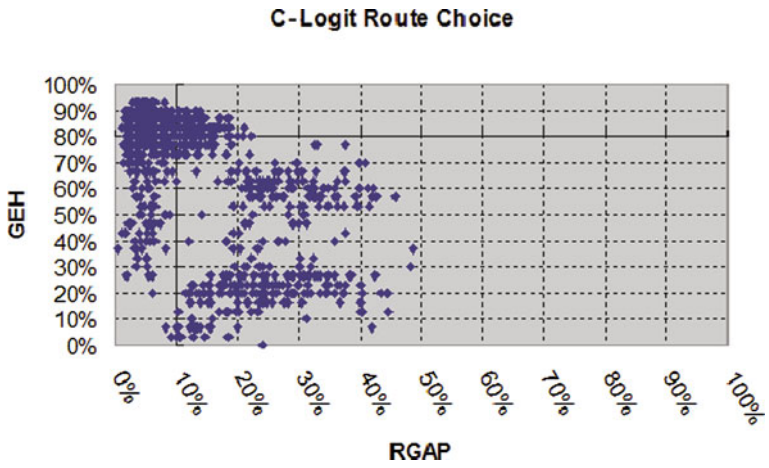


Fig. 1.15 Rgap and GEH for a heuristic dynamic assignment procedure based on the use of C-logit route choice model

close enough to the observed values. That means that at least two criteria have to be used in the calibration process to achieve acceptable results.

Independent of whether or not the objective function of equation (1.35) is based on one or more performance criteria, to solve the optimization problem (1.35), various methods have been used by researchers, usually based on standard optimization packages, such as MINOS (Hourdakis et al., 2003) and LINDO (Ciuffo et al., 2007). Genetic algorithms have also been used as optimization methods, which have attracted the attention of some researchers. (See, for instance, Ma and Abdulhai, 2002; Kim and Rilett, 2004; and Ma et al., 2006.)

However, the classical optimization methods, being deterministic in essence, do not explicitly take into account neither the inherent stochastic nature of traffic phenomena nor the role of interactions between model parameters. Consequently, it has been quite natural that the research community drew its attention to optimization methods that are explicitly designed for heuristically optimizing models of stochastic systems. A method that has already proved successful with simulation models is the simultaneous perturbation stochastic approximation (SPSA) algorithm developed by Spall (1998, 2003). This recursive optimization algorithm does not depend on direct gradient information of measurements. Rather, it depends on an approximation to the gradient formed from generally noisy measurements of the objective function. This objective function measures the model performance, which is an implicit function of the parameters to be calibrated. It does not require the detailed knowledge of the functional relationship between the parameters being adjusted and the objective function being minimized. The approaches, based on gradient approximations, require only conversion of the basic output measurements in order to sample values of the objective function, which does not require full knowledge of the system input–output relationships. The gradient approximation takes the form

$$\widehat{g}_k(\widehat{\theta}_k) = \frac{L(\widehat{\theta}_k + c_k\Delta_k) - L(\widehat{\theta}_k - c_k\Delta_k)}{2c_k}\left[\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, ..., \Delta_{kp}^{-1}\right]^{\mathrm{T}} \tag{1.37}$$

where Δ_{ki} is the ith component of the vector Δ_k of simultaneous perturbations, c_k is a positive scalar, and $L(\theta)$ is the objective function measuring the system's performance in terms of a continuous p-dimensional vector θ of the parameters on which the simulation model depends. The values of these parameters have to be determined to estimate the best value of $L(\theta)$. The objective function $L(\theta)$ can be any of the functions that have been referred to above. In what follows, for the sake of simplicity, we will take the RMSE function that measures the distances between the N observed flows y_i at each detection station in the network and the corresponding N simulated flows $x_i(\theta)$, whose values depend on the values of parameters in θ:

$$L(\theta) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[x_i(\theta) - y_i\right]^2} \tag{1.38}$$

The gradient approximations $g(\theta) = \partial L(\theta)/\partial\theta$ are built from noisy measurements of $L(\theta)$:

$$L\left(\hat{\theta}\right) = L(\theta) + \varepsilon$$

where ε is a random error function satisfying certain conditions (Spall, 2003). The basic unconstrained SPSA algorithm is in the general recursive form

$$\widehat{\theta}_{k+1} = \widehat{\theta}_k - a_k\widehat{g}_k\left(\widehat{\theta}_k\right) \tag{1.39}$$

where $\widehat{g}_k\left(\widehat{\theta}_k\right)$ is the simultaneous perturbation estimate of the gradient $g(\theta)$ at the iterate $\widehat{\theta}_k$ given by eq. (1.37) and based on the measurements of the objective function. The essential part of eq. (1.39) is the two-sided gradient approximation (1.37) formed by randomly perturbing the components of $\widehat{\theta}_k$ with the mean-zero random perturbation vector:

$$\Delta_k = \left[\Delta_{k1}, \Delta_{k2}, ..., \Delta_{kp}\right]^{\mathrm{T}} \tag{1.40}$$

This unconstrained version of the SPSA algorithm has to be suitably adapted for dealing with the upper and lower values bounding the range of feasible values of model parameters. Sadegh (1997) proves that if the set $G=\{\theta: q_i(\theta) \le 0, i=1,\ldots,p\}$ is non-empty and bounded, the functions $q_i(\theta)$, $i=1,..,p$ are continuously differentiable, and at each $\theta\in\partial G$, where ∂ denotes the boundary, the gradients of the active bounding constraints are linearly independent, then the algorithm converges to a Karush–Kuhn–Tucker optimal point. In this case, eq. (1.39) can be replaced by a projection algorithm

$$\widehat{\theta}_{k+1} = P\left[\widehat{\theta}_k - a_k \widehat{g}_k\left(\widehat{\theta}_k\right)\right] \tag{1.41}$$

that projects $\hat{\theta}_{k+1}$ onto the feasible region G. The main convergence conditions of the SPSA algorithm (Spall, 2003) require the following:

1. The scalar nonnegative gain coefficients $a_k > 0$ and $c_k > 0$ must satisfy
   For a_k and $c_k \to 0$: $\sum_{k=0}^{\infty} a_k = \infty$ and $\sum_{k=0}^{\infty} \frac{a_k^2}{c_k^2} < \infty$
2. Iterate boundedness condition: $\underset{k \geq 0}{\text{Sup}} \left\| \widehat{\theta}_k \right\| < \infty$
3. Measurement noise: the ratio of measurement to perturbation must be such that

$$E\left[\left(L\left(\widehat{\theta}_k \pm c_k \Delta_k\right)/\Delta_{ki}\right)^2\right] \tag{1.42}$$

   is uniformly bounded over k and i.
4. Statistical properties of the perturbations: the $\{\Delta_{ki}\}$ are independent of all k, i, identically distributed for all i at each k, symmetrically distributed about zero, and uniformly bounded in magnitude for all k, i.

Conditions 1, 3, and 4 govern the gains a_k, c_k, and the random perturbations Δ_k and the square summability in condition 1 balances the decay in a_k against c_k to ensure that the update in moving $\widehat{\theta}_k$ to $\widehat{\theta}_{k+1}$ is well behaved. Spall proves that an important and very simple distribution that makes perturbations Δ_k satisfy condition 3 is the symmetric Bernoulli ± 1 distribution.

The basic SPSA algorithm can be stated in the following form (Spall, 2003):

Step 0: Initialization and coefficient selection

Set counter index $k = 0$

Pick initial guess $\widehat{\theta}_0$ and nonnegative coefficients a, c, A, α, and γ in the SPSA gain sequences:

$$a_k = \frac{a}{(k+1+A)^{\alpha}} \text{ and } c_k = \frac{c}{(k+1)^{\gamma}} \tag{1.43}$$

Practically effective and theoretically valid values for α and γ are 0.602 and 0.101, respectively, a, c, and A may be determined based on practical guidelines, see Spall (2003) for details.

Step 1: Generation of the simultaneous perturbation vector

Generate by Monte Carlo a p-dimensional random perturbation vector Δ_k, where each of the p components of Δ_k is independently generated from a zero-mean probability distribution satisfying condition 3.

Step 2: Objective function evaluation

Obtain two measurements of the loss function based on the simultaneous perturbation around the current $\widehat{\theta}_k$

$$L\left(\widehat{\theta}_k + c_k\Delta_k\right) \text{ and } L\left(\widehat{\theta}_k - c_k\Delta_k\right)$$

with the current c_k, a_k, and Δ_k (i.e., run two simulation experiments with the corresponding perturbations of the values of the parameters to be calibrated).

Step 3: Gradient approximation

Generate the simultaneous perturbation approximation to the unknown gradient $g_k\left(\widehat{\theta}_k\right)$ according to

$$\widehat{g}_k\left(\widehat{\theta}_k\right) = \frac{L\left(\widehat{\theta}_k + c_k\Delta_k\right) - L\left(\widehat{\theta}_k - c_k\Delta_k\right)}{2c_k}\left[\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, ..., \Delta_{kp}^{-1}\right]^{\mathrm{T}}$$

Remark: it is sometimes useful to average several gradient approximations at $\widehat{\theta}_k$, each formed from an independent generation of Δ_k if the noise effects ϵ_k are relatively large (an empirical rule that has provided satisfactory results has been to average the gradient approximation from three independent generations of Δ_k).

Step 4: Update θ estimate

Using $\widehat{\theta}_{k+1} = P\left[\widehat{\theta}_k - a_k\widehat{g}_k\left(\widehat{\theta}_k\right)\right]$

Step 5: Termination test

If $\left|L\left(\widehat{\theta}_k\right) - L\left(\widehat{\theta}_{k+1}\right)\right| \leq \Phi$ (where Φ is an acceptable error bound), then stop. Otherwise set the iteration counter $k \leftarrow k+1$, update a_k and c_k, and repeat from Step 1.

Applications of SPSA for calibrating microsimulation parameters can be found in Balakrishna (2006), Balakrishna et al. (2007a, b), Ma et al. (2007), Barceló et al. (2007). Balakrishna's papers deal with the application in the context of calibrating OD matrices. In Ma et al., the method is successfully applied for calibrating global car-following parameter jointly with local parameters, as in Lee and Ozbay (2008), while Barceló et al. (2007) use it to calibrate the route choice parameters. Unfortunately there is no evidence yet that any method is clearly superior to all others. Ma et al. (2007) conclude that "compared with other heuristic methods such

as Genetic Algorithms, this method (SPSA) can generally obtain an acceptable set of parameters in much less time. Nonetheless, one cannot safely say one particular method would outperform all others in all cases".

So far, all calibration processes that have been discussed assume that the values of the model parameters are adjusted so that the model outputs fit the observed data acceptably well. This implicitly assumes that the observed data satisfies certain conditions of quality and reliability; few researchers have drawn their attention to the implications of the assumptions in the quality of the simulation (Bayarri et al., 2004). Under such conditions of quality and reliability, this assumption can be accepted for the simulation of systems in which route choice is not relevant. Therefore, it should not be surprising that most of the analyzed methods and their applications correspond to freeways and similar networks for which this is the case. Dealing with networks in which route choice is a key component – as is the case for urban networks, where vehicles travel from origins to destinations through flowing paths across the network – the calibration of route choice parameters, as part of the DTA component of the model, becomes a critical part of the calibration process, as has been underscored. However, the attempt to reproduce the observed data (measured flow, for instance) is a function not only of how well the model parameters have been calibrated (namely the route choice parameter) but of the quality and reliability of the input data defined in terms of an origin–destination matrix, preferably time sliced to account properly for the time variability of traffic demand. The assumption that OD matrices represent a fixed input, whose quality is taken for granted, usually yields contradictory situations. This is especially true given that the observable data (i.e., flow measurements for a given time period) are quite frequently available for a specific day, let us say a working Tuesday. Meanwhile, the only available OD matrix is the OD matrix for an "average working day," which is an abstraction that cannot be observed and does not correspond to a physical reality.

The first methodological consideration is that one should not expect that an "average working day OD matrix," which is an artificial and not an observable construction, succeeds in reproducing acceptably well the flow measurements of a specific day at given locations during a given time interval. A way of overcoming this inconsistency could be to adjust the given OD matrix from the measured values; this current practice in most applications of microscopic simulation is an attempt to ensure that there is a correspondence between traffic demand and the flows it has to reproduce. A methodologically sound process requires using independent measurement samples for each process, the adjustment of the OD matrix, and the calibration. Leaving aside the heuristic approaches based on static assignments for manipulating the original OD matrices in order to time slice them and adjust them to observed flows, a quick and dirty heuristic is better than nothing, but clearly inappropriate for capturing the dynamics of the phenomenon. The most significant approaches (Cascetta et al., 1993; Ashok and Ben-Akiva, 2000) are based on estimators that rely on the availability of an assignment matrix, which captures the interdependence of route choice and traffic dynamics and which also assumes that the parameters of such an assignment are given.

A more reliable approach seems to be joint estimation and parameter calibration, that is, make the adjustment of the OD matrix to the available measurement part of the model calibration and use a different independent sample of observed values for validation purposes. Toledo and Koutsopoulos (2004) present a proposal that decomposes the calibration process into two phases: an aggregated part where car-following and lane-changing parameters are calibrated on the basis of disaggregate measurements (i.e., vehicle trajectories), followed by an iterative procedure based on aggregate measurement to jointly calibrate the OD matrix and the OD flows. From a theoretical standpoint, this approach is quite sound. However, it is unclear how it can be used by practitioners in the usual projects, in which microsimulation is used, given that such disaggregate date is unlikely to be available. More practical insight could come from the improved aggregated approach of Balakrishna et al. (2007b).

The proposed procedure divides the time horizon of interest [0,*T*], into a number of intervals of length Δt. If $f_{rsp}(t)$ is the flow on path p from origin r to destination s departing origin r at time interval t, $\mathbf{f_t}$ is the vector of flows departing their respective origins at time t, $t = 1,2,\ldots,n$, $n = T/\Delta t$, and β_t is the vector of model parameters to be calibrated together with the OD flows, then the calibration problem can be formulated in terms of the following optimization framework:

$$\text{Min } \mathbf{F}\left(\mathbf{f_1},...,\mathbf{f_n},\boldsymbol{\beta}_\mathbf{1},...,\boldsymbol{\beta}_\mathbf{n}\right) = \sum_{t=1}^{n}\left[F_1\left(M_t^{\text{obs}} - M_t^{\text{sim}}\right) + F_2\left(\mathbf{f_t},\mathbf{f_t^0}\right) + F_3\left(\boldsymbol{\beta}_\mathbf{t},\boldsymbol{\beta}_\mathbf{t}^\mathbf{0}\right)\right]$$

subject to :

$$L_t^f \le \mathbf{f_t} \le U_t^f, t = 1,2,...,n$$

$$L_t^\beta \le \boldsymbol{\beta}_\mathbf{t} \le U_t^\beta, t = 1,2,...,n$$

where $L_t^f, U_t^f, L_t^\beta, U_t^\beta$ are the lower and upper bounds, respectively, on OD flows and model parameters, $\mathbf{f_t^0}, \boldsymbol{\beta}_\mathbf{t}^\mathbf{0}$ are a priori values for OD flows and model parameters, and M_t^{obs} and M_t^{sim} are the observed and simulated traffic measurements. M_t^{sim} is the output of the simulation model that can be considered as a function $\phi(.)$ of the simulation model, the model parameters β_t, and the network G_t for interval t is $M_t^{\text{sim}} = \varphi\left(\mathbf{f_1},...,\mathbf{f_n},\boldsymbol{\beta}_\mathbf{1},...,\boldsymbol{\beta}_\mathbf{n}\right)$, assuming that the network does not change during the simulation horizon.

The function ϕ does not have an analytical form. Typically it is approximated by a linear mapping between OD flows and link counts, depending on an assignment matrix which is estimated by tracking vehicle trajectories (Ashok and Ben-Akiva, 2000). The proposal of Balakrishna (2006) and Balakrishna et al. (2007a, b) has the advantage of directly capturing the assignment matrix using the simulation model as a black box.

The functions F_1, F_2, and F_3 in the objective function $\mathbf{F}$ can be any of the goodness-of-fit functions discussed in this section, although Toledo et al. (2004) also propose functions of the form

$$Z = \left(M^{\text{obs}} - M^{\text{sim}}\right)^{\text{T}} W_1^{-1} \left(M^{\text{obs}} - M^{\text{sim}}\right) + \left(\mathbf{f} - \mathbf{f^o}\right)^{\text{T}} W_2^{-1} \left(\mathbf{f} - \mathbf{f^o}\right) + \left(\boldsymbol{\beta} - \boldsymbol{\beta^o}\right)^{\text{T}} W_3^{-1} \left(\boldsymbol{\beta} - \boldsymbol{\beta}^{\text{o}}\right)$$

where W_1, W_2, and W_3 are variance–covariance matrices of the traffic measurements, OD values, and behavior parameters, respectively.

The optimization problem can be solved iteratively by the SPSA method.

1.6 Concluding Remarks

Traffic simulation is a model-building and analysis technique useful for assisting professionals in a wide variety of traffic and transportation studies, with an even wider variety of objectives, ranging from assistance in designing new systems to evaluating impacts of alternative designs under different conditions. Traffic simulation is also becoming a key instrument in the design and evaluation of intelligent transport systems, especially when real-time management operations are a critical aspect of the system. There is a key reason for that. Dealing with time dependencies of traffic phenomena is a must for real-time systems and, so far, traffic simulation models are the most suitable tools for properly dealing with the variability of traffic over time. Traffic simulation is also a valuable tool for researchers to get a better understanding of traffic phenomena and to build virtual laboratories where they may conduct experiments for testing hypotheses.

This introductory chapter had several objectives:

- Provide an insight into the model-building principles and at the same time make evident that models are not independent of the problems they have to solve. Therefore, it is critical for any successful application of models to have a clear definition of the problem to solve, to identify the model that suits better the type of problem, and check the availability of the data required to build and apply the model properly.
- Introduce the reader to the main traffic models, the computation engines of the simulation software, their modeling hypothesis, and their consequences. A good understanding of the advantages and disadvantages of each of them should pave the way for a good practice in suing the models.
- Make the professional as well as the researcher aware that the existing models and the software implementing them do not represent the end of path. Although many problems have been solved in recent years, there are yet many others waiting for solution. This chapter has modestly underscored some of the most important.
- And finally, recalling Humpty Dumpty's assertion, make clear that in spite of the analyst's beliefs embedded into the model, neither models mean just what the analyst chooses to mean nor they can make the model mean many different thinks. Calibration and validation are the exercises to make the model mean just what they have to mean.

But this task also calls the analyst for a sound perception of the reality and not confuse wind mills with giants, as Don Quixote did.

References

Abou-Rahme N, White J (1999) Replicating Shockwaves within a classical car-following model. In: Transport Research Laboratory, Crowthorne, UK, Annual Research Review, pp. 73–82

Ackoff RL (1962) Scientific Method: optimizing applied research decisions, John Wiley & Sons, Inc., New York

Ahmed KI (1999) Modeling drivers' acceleration and lane changing behaviors. PhD thesis, Massachusetts Institute of Technology

Ashok K, Ben-Akiva M (2000) Alternative approaches for real-time estimation and prediction of time-dependent origin-destination flows, Transportation Science, vol.34, (1), pp. 21–36

Aycin MF, Benekohal RF (1998) Linear acceleration carfollowing model development and validation. Transportation research record no. 1644, pp 10–19

Balakrishna R (2006) Off-line calibration of dynamic traffic assignment models. PhD thesis, Massachusetts Institute of Technology

Balakrishna B, Antoniou C, Ben-Akiva M, Koutsopoulos H, Wen Y (2007a) Calibration of microscopic traffic simulation models: methods and application, TRR no. 1999, pp 198–206

Balakrishna R, Ben-Akiva M, Koutsopoulos HN (2007b) Off-line calibration of dynamic traffic assignment: simultaneous demand–supply estimation. Proceedings of the 86th annual meeting of the transportation research board, Washington, DC

Barceló J, Casas J (2004) Methodological notes on the calibration and validation of microscopic traffic simulation models. Paper 04-4975 presented at the 83rd transportation research board annual meeting, Washington, DC

Barceló J, Casas J (2006) Stochastic heuristic dynamic assignment based on AIMSUN microscopic traffic simulator. Paper #06-3107 presented at 85th transportation research board annual meeting, Transportation research records no. 1964, pp 70–79

Barceló J, Perarnau J, Torday A (2007) Automated calibration assistance tools for microscopic traffic simulators. Presented at the world conference on transportation research, included in the CD conference proceedings, Berkeley

Bayarri MJ, Berger JO, Higdon D, Kennedy MC, Kottas A, Paulo R, Sacks J, Cafeo A, Cavendish JC, Lin CH, Tui J (2002) A framework for validation of computer models. NISS technical report number 128

Bayarri MJ, Berger JO, Molina G, Rouphail NM, Sacks J (2004) Assessing uncertainties in traffic simulation: a key component in model calibration and validation. Paper #04-2504, 83rd TRB meeting, Washington, DC

Ben-Akiva M, Bierlaire M (1999) Discrete Choice Methods and Their Applications to Short Term Travel Decisions, In: Hall RW (ed) Handbook of Transportation Science, Kluwer's International Series, Kluwer Academic Publishers, The Netherlands

Ben-Akiva M, Lerman SR (1985) Discrete choice analysis: theory and applications to travel demand. MIT Press, Cambridge, Massachussets

Ben-Akiva M, Bierlaire M, Bottom J, Koutsopoulos HN, Mishalani RG (1997) Development of a route guidance generation system for real-time application. Proceedings of the 8th IFAC symposium on transportation systems, Chania, Crete

Ben-Akiva M, Bierlaire M, Burton D, Kotsopoulos HN, Mishalani R (2001) Network state estimation and prediction for real-time traffic management. Netw Spatial Econ 1:293–318

Ben-Akiva M, Bierlaire M, Koutsopoulos HN, Mishalani R (2002) Real-time simulation of traffic demand–supply interactions within DynaMIT. In: Gendreau M, Marcotte P (eds) Transportation and network analysis: current trends. Miscellanea in honour of Michael Florian. Kluwer, Boston

Benekohal RF, Treiterer J (1988) CARSIM: car-following model for simulation of traffic in normal and stop-and-go conditions. Transportation research record No. 1194, pp 99–111

Burghout W (2004) Hybrid microscopic–mesoscopic traffic simulation. Doctoral thesis, Royal Institute of technology, Stockholm, Sweden

Burghout W, Kotsopoulos H, Andréasson I (2005) Hybrid mesoscopic–microscopic traffic simulation. Proceedings of the 83rd TRB annual meeting, Washington, DC

Carey M, Ge YE (2007) Comparison of methods for path flow reassignment for dynamic user equilibrium. School of Management & Economics, Queen's University, Belfast, Northern Ireland

Cascetta E (2001) Transportation systems engineering: theory and methods. Kluwer, Academic Publishers. The Netherlands

Cascetta E, Inaudi D, Marquis G (1993) Dynamic estimators of origin–destination matrices using traffic counts. Transport Sci 27(4):363–373

Cascetta E, Nuzzolo A, Russo F, Vitetta A (1996) A modified logit route choice model overcoming path overlapping problems. In: Proceedings of the 13th international symposium on transportation and traffic flow theory. Pergamon, Elsevier Science Ltd., Oxford, UK

Chandler FE, Herman R, Montroll EW (1958) Traffic dynamics: studies in car following. Oper Res 6:165–184

Chen S, Sheridan TB, Kusunoki H, Komoda N (1995) Car-following measurements, simulations, and a proposal procedure for evaluating safety. Analysis, design and evaluation of man–machine systems, vol. 2. Pergamon Press, pp. 529–534

Choudhury C (2007) Modeling driving decisions with latent plans. PhD thesis, Massachusetts Institute of Technology

Choudhury C, Ben-Akiva M, Toledo T, Lee G, Rao A (2007) Modeling cooperative lane-changing and forced merging behavior. Proceedings of the 86th transportation research board annual meeting, Washington, DC

Choudhury C, Ramanujam V, Ben-Akiva M (2008) A lane changing model for urban arterials. Proceedings of the 3rd international symposium of transport simulation, Gold Coast, Australia

Chronopoulos AT, Michalopoulos P, Donohoe J (1992) Efficient traffic flow simulation computations. Math Comput Model 16(5):107–120

Ciuffo B, Punzo V, Torrieri V (2007) A framework for calibrating microscopic traffic simulation models. Proceedings of the 86th TRB annual meeting, Washington, DC

Daellenbach HG (1995) Systems and decision making: a management science approach. John Wiley & Sons Ltd., Chichester, England

Daganzo CF (1994) The cell-transmission model: a simple dynamic representation of highway traffic. Transport Res 288(4):269–281

Daganzo CF (1997) Fundamentals of transportation and traffic operations. Pergamon, Elsevier Science Ltd., Oxford, UK

Del Castillo JM (1994) A theory of car following. PhD Thesis, Universidad de Sevilla, Spain

Del Castillo JM, Benitez FG (1995) On the functional form of the speed–density relationship I: general theory. Transport Res 29B(5):373–389

Dowling R, Skabardonis A, Halkias J, McHaie G, Zammit G (2004) Guidelines for calibration of microsimulation models: framework and application. In: Transportation research records: Journal of the Transportation Research Board, No. 1876, TRB, National Research Council, Washington, DC

Edie LC (1960) Car following and steady theory for non-congested traffic flow. Oper Res 9:66–76

Fellendorf M (1994) VISSIM: a microscopic simulation tool to evaluate actuated signal control including bus priority. Technical paper, Session 32, 64th ITE annual meeting, Dallas, TX

FHWA (2004) Traffic analysis toolbox, volume II: guidelines for applying traffic microsimulation software, Publication FHWA-HRT-04-040, FHWA, US Department of Transportation, http://ops.fhwa.dot.gov/trafficanalysistools/tat_vol3

Florian M, Hearn D (1995) Network equilibrium models and algorithms. In: Ball MO et al. (eds) Handbooks in OR & MS, vol. 8. Elsevier, Amsterdam

Florian M, Mahut M, Tremblay N (2001) A hybrid optimization–mesoscopic simulation dynamic traffic assignment model. Proceedings of the 2001 IEEE intelligent transport systems conference, Oakland, pp 118–123

Florian M, Mahut M, Tremblay N (2002) Application of a simulation-based dynamic traffic assignment model. Presented at the international symposium on transport simulation, Yokohama (also in: Kitamura R, Kuwahara M (eds) Simulation approaches in transportation analysis. Kluwer, 2005, pp 1–21)

Fox K (2008) Is Micro-Simulation a waste of Time? Proceedings of the European Transport Conference, Leeuwenhorst Conference Centre, The Netherlands

Friesz TL, Bernstein D, Smith TE, Tobin RL, Wie BW (1993) A variational inequality formulation of the dynamic network user equilibrium problem. Oper Res 41(1):179–191

Gazis DC, Herman R, Potts RB (1959) Car-following theory of steady state flow. Oper Res 7(4):499–505

Gazis DC, Herman R, Rothery RW (1961) Mon-linear follow-the-leader models of traffic flow, Oper. Res. 9 (4), pp. 545–567

Gerlough DL, Huber MJ (1975) Traffic flow theory: a monograph. TRB special report 165

Gipps PG (1981) A behavioral car-following model for computer simulation. Transport Res B 15:105–111

Greenberg H (1959) An analysis of traffic flow. Oper Res 7(1):79–85

Greenshields BD (1934) A study of traffic capacity. HRB Proc 14:448–477

Herman R, Rothery RW (1965) Car following and steady state flow. In: Almond J (ed) Proceedings of the 2nd international symposium on the theory of traffic flow. OECD, Paris

Herman R, Montroll EW, Potts RB, Rothery RW (1959) Traffic dynamics: analysis of stability in car following. Oper Res 7(1):86–106

Hidas P (1998) A car following model for urban traffic simulation. Traffic Eng Control 39(5): 300–305

Hidas P (2002) Modelling lane changing and merging in microscopic traffic simulation. Transport Res C Emerg Technol 10(5–6):351–371

Hidas P (2004) Evaluation of lane changing and merging in microsimulation models. Papers of the 27th Australasian transport research forum, Adelaide

Highway Agency (1996) Design manual for roads and bridges, vol. 12, Traffic appraisal of road schemes, Section 2.5, Part I, Traffic appraisal in urban areas. The Stationery Office, London

Hirsch C (1988) Numerical Computation of Internal and External Flows, vol. 1, Wiley Series in Numerical Method in Engineering, John Wiley and Sons, Chichester, England

Hollander Y, Ronghui L (2008) The principles of calibrating traffic microsimulation models. Transportation 35:347–362

Hourdakis J, Michalopoulos PG, Kottommannil J (2003) A practical procedure for calibrating microscopic traffic simulation models. Presented at the 82nd TRB annual meeting, Washington, DC

Janson BN (1991) Dynamic assignment for urban road networks. Transport Res B 25(2/3):143–161

Jayakrisham R, Mahmassani HS, Yu TY (1994) An evaluation tool for advanced traffic information and management systems in urban networks. Transport Res C 2C(3):129–147

Kim K, Rilett LR (2004) A genetic algorithm based approach to traffic micro-simulation calibration using ITS data. 83rd TRB annual meeting, Washington D.C.

Kühne R (1989) Microscopic distance strategies and macroscopic traffic flow models. In: CCCT'89, Control, computers, communication in transportation. AFCET, Paris, pp 267–273

Kühne R, Michalopoulos P, Zhang HM (1992) Continuum flow models, Chapter 5 in Traffic flow theory, http://www.tfhrc.gov

Law AM, Kelton WD (1991) Simulation modeling and analysis. McGraw-Hill, International Editions, New York

Leonard DP, Gower P, Taylor N (1989) CONTRAM. Structure of the model. Transport and Road Research Laboratory, Research report 178, Department of Transport, Crowthorne

Leutzbach W (1988) Introduction to the theory of traffic flow. Springer-Verlag, Berlin

Liu R, Hyman G (2008) Generic guidance for modeling merges. Proceedings of the European transport conference, Leeuwenhorst Conference Centre, The Netherlands

Liu HX, Ma W, Ban JX, Mirchandani P (2005) Dynamic equilibrium assignment with microscopic traffic simulation. Paper presented at the 8th international IEEE conference on intelligent transportation systems, Wien

Lo HK, Szeto WY (2002) A cell-based variational inequality formulation of the dynamic user optimal assignment problem. Transport Res B 36:421–443

Lubashevsky I, Wagner P, Mahnke R (2003) Rational-driver approximation in car-following theory. Phys Rev E 68, pp. 1–15

Ma T, Abdulhai B (2002) A genetic algorithm-based optimization approach and generic tool for calibrating traffic microscopic simulation parameters. Transportation research record 1806, pp 6–15

Ma J, Dong H, Zhang M (2006) Calibration of departure time and route choice parameters in micro-simulation with macro measurements and genetic algorithm. 85th TRB annual meeting, paper no. 2376, Washington D.C.

Ma J, Dong H, Zhang M (2007) Calibration of microsimulation with heuristic optimization methods. TRR no. 1999, pp 208–217

Mahmassani H (2001) Dynamic network traffic assignment and simulation methodology for advanced system management applications. Netw Spatial Econ 1:267–292

Mahmassani HS, Hu TY, Peeta S, Ziliaskopoulos A (1994) Development and testing of dynamic traffic assignment and simulation procedures for ATIS/ATMS applications. Technical report DTFH61-90-R00074-FG, Center for Transportation Research, The University of Texas at Austin

Mahut M (1999) Behavioural car following models. Report CRT-99-31, Centre for Research on Transportation, University of Montreal, Montreal, Canada

Mahut M (2000) Discrete flow model for dynamic network loading. PhD thesis, Département d'Informatique et de Recherhe Opérationelle, Université de Montréal. Published by the Center for Research on Transportation (CRT), University of Montreal

Mahut M, Florian M, Tremblay N (2003a) Space–time queues and dynamic traffic assignment: a model, algorithm and applications. Transportation Research Board, 82nd annual meeting, Washington D.C.

Mahut M, Florian M, Tremblay N (2003b) Traffic simulation and dynamic assignment for off-line applications. Presented at the 10th world congress on intelligent transportation systems, Madrid

Mahut M, Florian M, Tremblay N (2004) INRO Consultants Inc., Montréal, QC and M. Campbell, D. Patman and Z. Krnic McDaniel, City of Calgary, Calgary, AB. Calibration and application of a simulation based dynamic traffic assignment model, Transportation research record 1876, pp 101–111. Available at http://www.inro.ca/en/community/pres_pap/index.php#204

Mahut M, Florian M, Tremblay N (2008) Comparison of assignment methods for simulation-based dynamic-equilibrium traffic assignment. Presented at the 88th annual meeting of the transportation research board, Washington, DC

May A (1990) Traffic flow fundamentals. Prentice Hall, Englewood Cliffs, New Jersey

May AD, Keller HEM (1967) Non-integer car-Following Models, Highw. Res. Rec. 199, pp. 19–32

Messmer A, Papageorgiou M (1990) METANET: a macroscopic simulation program for motorway networks. Traffic Eng Control 31:466–470

Michalopoulos PG (1988) Analysis of traffic flows at complex congested arterials. Transport Res Rec 1194:77–86

Michalopoulos PG, Yi P, Beskos DE, Lyrintzis AS (1991) Continuum modeling of traffic dynamics. In: Proceedings of the 2nd International Conference on Application of Advanced Technology in Transportation Engineering, August 18–21, Minneapolis, Minnesota, pp 36–40

Mitchell G (1993) The practice of operations research. John Wiley & Sons, Chichester, England

Newell GF(2002) A simplified car-following theory: a lower order model. Transport Res B 36: 195–205

NGSIM (2002) Next generation simulation program. Federal Highway Administration (FHWA), www.ngsim.fhwa.dot.gov
OECD 1987) Dynamic traffic management in urban and suburban road systems. Organization for Economic Co-operation and Development, Paris
Panwai S, Dia H (2005) Comparative evaluation of microscopic car-following behavior. IEEE Trans Intell Transport Syst 6(4):314–325
Papageorgiou M (1990) Dynamic modeling, assignment, and route guidance in traffic networks. Transport Res B 24B(6):471–495
Papageorgiou M, Schmidt G (1991) Freeway traffic modeling. In: Papageorgiou M (ed) Concise encyclopaedia of traffic and transportation systems. Pergamon Press, Elsevier Science Ltd., Oxford, UK
Papageorgiou M, Blosseville JM, Haj-Salem H (1989) Macroscopic modeling of traffic flow on the Boulevard Periphérique in Paris. Transport Res 23B:29–47
Papageorgiou M, Blosseville JM, Haj-Salem H (1990a) Modeling and real-time control of traffic flow on the southern part of Boulevard Périphérique in Paris, Part I. Model Transport Res 24A:345–359
Papageorgiou M, Blosseville JM, Haj-Salem H (1990b) Modeling and real-time control of traffic flow on the southern part of Boulevard Périphérique in Paris, Part II: Coordinated on-ramp metering. Transport Res 24A:361–370
Parker MT (1996) The effect of heavy goods vehicles and follower behavior on capacity at motorway roadwork sites. Traffic Eng Control 37(9):524–531
Payne HJ (1971) Models of freeway traffic and control. Simul Council Proc 1:51–61
Payne HJ (1979) FREEFLO: a macroscopic simulation model of freeway traffic. Transport Res Rec 772:68–75
Peeta S, Mahmassani, HS (1995) System optimal and user equilibrium time-dependent traffic assignment in congested networks. Ann Oper Res 60:81–113
Pipes LA (1953) An operational analysis of traffic dynamics. J Appl Phys 24(3):274–287
Punzo V, Tripodi A (2007) Steady-state solutions and multiclass calibration of Gipps microscopic traffic flow model. TRR no. 1999, pp 104–114
Rakha H, Gao Y (2009) Calibration of steady-state car-following models using macroscopic loop detector data. Paper presented at the 88th TRB annual meeting, Washington, DC
Rakha H, Pecker CC, Cybis HBB (2007) Calibration procedure for Gipps car-following model. TRR no. 1999, pp 115–127
Ran B, Boyce D (1996) Modeling dynamic transportation networks. Springer, Lecture Notes on Transportation, Berlin
Ranney TA (1999) Physiological factors that influence car-following and car-following model development. Transport Res F Traffic Psychol Behav 2:213–219
Reuschel A (1950a) Vehicle movements in a platoon, Osterreichisches Ing.-Arch 4:193–215
Reuschel A (1950b) Vehicle movements in a platoon with uniform acceleration or deceleration of the lead vehicle. Zeitschrift des Osterreichischen Ingenieur und Architekten Vereines (95): 59–62, 73–77
Ross P (1988) Traffic dynamics. Transport Res B 22B(6):421–435
Rothery RW (1992) Car following models, Chapter 4, in Traffic flow theory, http://www.tfhrc.gov
Rouphail NM, Sacks J (2003) Thoughts on traffic models calibration and validation. Paper presented at the workshop on traffic modeling, Sitges, Spain
Sadegh P (1997) Constrained optimization via stochastic approximation with a simultaneous perturbation gradient approximation. Automatica 33:889–892
Sbayti H, Lu C, Mahmassani HS (2007) Efficient implementations of the method of successive averages in simulation-based DTA models for large-scale network applications. TRB 2007 annual meeting. Washington, DC
Sheffi Y (1985) Urban transportation networks: equilibrium analysis with mathematical programming networks. Prentice-Hall, Inc., Englewood Cliffs, New Jersey

Spall JC (1998) An overview of the simultaneous perturbation method for efficient optimization. John Hopkins APL Tech Digest 19(4):482–492
Spall JC (2003) Introduction to stochastic search and optimization: estimation, simulation and control. Wiley-Interscience, John Wiley & Sons Inc., New Jersey
Stewart D (2003) Calibrating the freeway merge in Aimsun2 for the pacific motorway project, Queensland Main Roads, Traffic and Road Use Management Division
Theil H (1961) Economic forecasts and policy. North-Holland, Publishing Company, Amsterdam
Toledo T, Koutsopoulos H (2004) Statistical validation of traffic simulation models. Proceedings of the 82nd annual meeting of the transportation research board, Washington, DC
Tong CO, Wong SC (2000) A predictive dynamic traffic assignment model in congested capacity-constrained road networks. Transport Res B 34:625–644
Varia HR, Dhingra SL (2004) Dynamic user equilibrium traffic assignment on congested multidestination network. J Transport Eng 130(2):211–221
Wagner P, Lubashevsky I (2006) Empirical basis for car-following theory development. Research Report Institute of Transport Research, German Aerospace Center, Germany
Wang J, Ronghui L, Montgomery F (2005) Paper presented at the 84th annual meeting of the transportation research board, Washington, BC
Wardrop JG (1952) Some theoretical aspects of road traffic research. Proceedings of the Institute of Civil Engineers, Part II, pp 325–378
Wiedemann R (1974) Simulation des Verkehrsflusses. Schriftenreihe des Instituts für Verkehrwessen, Heft 8, Universität (TH) Karlsruhe
Wilson RE (2001) An analysis of Gipps' car-following model of highway traffic. IMA J Appl Math 66:509–537
Wu JH (1991) A study of monotone variational inequalities and their application to network equilibrium problems. Ph.D. thesis, Centre de Recherche sur les Transports, Université de Montréal, Publication #801
Wu JH, Chen Y, Florian M (1998a) The continuous dynamic network loading problem: a mathematical formulation and solution method. Transport Res B 32(3):173–187
Wu JH, Florian M, Xu YW, Rubio-Ardanaz JM (1998b) A projection algorithm for the dynamic network equilibrium problem. In: Yang Z, Wang KCP, Mao B (eds) Traffic and transportation studies, Proceedings of the ICTTS'98. ASCE, pp 379–390
Xu YW, Wu JH, Florian M (1998) An Efficient Algorithm for the Continuous Network Loading Problem: a DYNALOAD Implementation. In: Bell MGH (ed) Transportation Networks: Recent Methodological Advances, Pergamon Press, Elsevier Science Ltd. Oxford, UK
Xu YW, Wu JH, Florian M, Marcotte P, Zhu DL (1999) Advances in the continuous dynamic network problem. Transport Sci 33(4):341–353
Yeo H, Skabardonis A (2007) NGSIM TO9: oversaturated freeway flow algorithm, http://www.tfhrc.gov
Yoshii T (1999) Standard verification process for traffic simulation model – verification manual. Kochi University of Technology, Kochi, Japan
Zhang M, Ma J (2008) Developing calibration tools for microscopic traffic simulation final report, Parts 1, 2 and 3, California PATH working papers UCB-ITS-PWP-2008-3

Chapter 2
Microscopic Traffic Flow Simulator VISSIM

Martin Fellendorf and Peter Vortisch

Traffic simulation is an indispensable instrument for transport planners and traffic engineers. VISSIM is a microscopic, behavior-based multi-purpose traffic simulation to analyze and optimize traffic flows. It offers a wide variety of urban and highway applications, integrating public and private transportation. Complex traffic conditions are visualized in high level of detail supported by realistic traffic models. This chapter starts with a review of typical applications and is followed by modeling principles presenting the overall architecture of the simulator. The Section 2.3 is devoted to core traffic flow models consisting of longitudinal and lateral movements of vehicles on multi-lane streets, a conflict resolution model at areas with overlapping trajectories and the social force model applied to pedestrians. The routing of vehicles and dynamic assignment will be described thereafter. Section 2.5 will present some techniques to calibrate the core traffic flow models. This chapter closes with remarks of interfacing VISSIM with other tools.

2.1 History and Applications of VISSIM

This section will familiarize you with some typical and some rather extraordinary studies being conducted by applying VISSIM. The examples presented will give you a flavor of the functionality and versatility of this microscopic traffic flow model embedded within a graphical user interface enabling traffic engineers without dedicated computer knowledge to set up microscopic traffic flow models.

VISSIM is a commercial software tool with about 7000 licenses distributed worldwide in the last 15 years. About one-third of the users are within consultancies

M. Fellendorf (✉)
University of Technology Graz, Rechbauerstrasse 12, Graz 8010, Austria
e-mail: martin.fellendorf@tugraz.at

P. Vortisch
Karlsruhe Institute of Technology, Institute for Transport Studies (IfV), Karlsruhe 76131, Germany
e-mail: peter.vortisch@kit.edu

J. Barceló (ed.), *Fundamentals of Traffic Simulation*,
International Series in Operations Research & Management Science 145,
DOI 10.1007/978-1-4419-6142-6_2, © Springer Science+Business Media, LLC 2010

and industry, one-third within public agencies, and the remaining third is applied at academic institutions for teaching and research. Primarily the software is suited for traffic engineers. However, as transport planning is looking toward a greater level of detail, an increasing number of transport planners use microsimulation as well.

VISSIM has a long lasting history; major steps within the development are presented in Table 2.1.

VISSIM is a microscopic, discrete traffic simulation system modeling motorway traffic as well as urban traffic operations. Based on several mathematical models described in Chapter 3, the position of each vehicle is recalculated every 0.1–1 s. The system can be used to investigate private and public transport as well as in particular pedestrian movements. Traffic engineers and transport planners assemble applications by selecting appropriate objects from a variety of primary building blocks. In order to simulate multi-modal traffic flows, technical features of pedestrians, bicyclists, motorcycles, cars, trucks, buses, trams, light (LRT), and heavy rail are provided with options of customization. Common applications include the following:

- Corridor studies on heavily utilized motorways to identify system performance, bottlenecks, and potentials of improvement.
- Advanced motorway studies including control issues like contra-flow systems, variable speed limits, ramp metering, and route guidance.
- Development and analysis of management strategies on motorways including mainline operation and operational impacts during phases of construction.
- Corridor studies on arterials with signalized and non-signalized intersections.
- Analysis of alternative actuated and adaptive signal control strategies in sub-area networks. Tools suited for traffic control optimization such as Linsig, P2, Synchro, or Transyt optimize cycle times and green splits for fixed-time control, while signal controllers in many countries are operated in a traffic responsive mode. Microsimulation is widely used for detailed testing of control logics and performance analysis.
- Signal priority schemes for public transport within multi-modal studies. Traffic circulation, public transport operations, pedestrian crossings, and bicycle facilities are modeled for various layouts of the street network and different options of vehicle detection.
- Alignment of public transport lines with various types of vehicles such as Light Rail Transit (LRT), trams, and buses with refinements in design and operational strategy. This also includes operation and capacity analysis of tram and bus terminals.
- Investigations on traffic calming schemes including detailed studies on speeds during maneuvers with limited visibility.
- Presentation of alternative options of traffic operations on motorways and urban environments for public hearings.

Table 2.1 Major steps of the development of VISSIM

Wiedemann (1974)	Secondary PhD thesis presenting the psycho-physical car-following model which describes the movement of vehicles on a single lane without exits
1978–1983	Several research projects at the university of Karlsruhe conducting measurements and model developments of particular pieces of vehicular movements. In particular Sparmann (1978) described lane changes on two-lane motorways, Winzer (1980) measured desired speeds on German motorways, Brannolte (1980) looked at traffic flow at gradients, and Busch and Leutzbach (1983) investigated lane-changing maneuvers on three-lane motorways
Hubschneider (1983)	PhD thesis on the development of a simulation environment to model vehicles on multi-lane streets and across signalized and non-signalized intersections. The model being called MISSION was implemented using SIMULA-67 and compiled on mainframe computer UNIVAC 1108
1983–1991	Research projects at the University of Karlsruhe applying MISSION for various studies on capacity and safety. Major applications contain noise (Haas, 1985) and emission (Benz, 1985) calculations, while Wiedemann and Schnittger (1990) looked at the impact of safety regulations on traffic flow. During that time MISSION had been installed under MS-DOS after reimplementation using PASCAL and MODULA-2
1990–1994	Wiedemann and Reiter (Reiter, 1994) recalibrated the original car-following model using an instrumented vehicle to measure the action points
Fellendorf (1994)	First commercial version of VISSIM in German aimed for capacity analysis at signalized intersections with actuated control. Graphical network editing, vehicle animation, and background maps were available from start on. The software was implemented in C running under MS-Windows 3.1
1994–1997	Rapid developments included definition of routes, further public transport modeling, modeling of priority intersections, and interfaces to various signal controller firmware
1998	Additional traffic flow model reducing the complexity of the original traffic flow model, which is better suited to calibrate traffic conditions on dense motorways. Further graphical enhancements like 3D-visualization
2000	Introduction of dynamic assignment as applications become larger and definition of routes are too time consuming
2003	COM interface providing users a standardized application programming interface to develop specific user applications with VISSIM in the background
2004	Interface between the strategic demand model VISUM and VISSIM to generate applications based on the same network geometry and traffic flow data
2006	Introduction of an implementation using multi-processor and distributed PC clusters for parallel processing to decrease computational time for large networks
2007	Anticipated driving at conflict areas
2008	Pedestrian modeling based on Helbing and Molnár (1995) social force model

2.2 Model Building Principles

For clarity the following definitions will be made. A microscopic traffic flow *simulator* such as VISSIM contains the software including the mathematical models to run traffic flow models. The simulator itself does not include any application-specific data or additional tools which are required for additional modeling and data analysis tasks. Additional software tools are sometimes required for data analysis such as statistical packages, proprietary external traffic control software, or post-processing such as emission calculations. These additional tools are not part of the simulator itself. A microscopic traffic *modeling application* contains all data to run a VISSIM model without the simulator itself. In order to operate an application, a modeling system is required. The microscopic *modeling system* comprises the simulator, all additional tools needed to operate an application and the data of a particular application.

The VISSIM software is implemented in C++ considering guidelines of object-oriented programming (OOP). Actually the OOP concept has first been introduced using ship simulation. The academic implementation and predecessor of VISSIM, MISSION of the University of Karlsruhe, was implemented using SIMULA-67 featuring classes of objects, virtual methods, and coroutines. VISSIM provides classes of objects such as vehicles. Within a class the properties of each object are characterized by attribute values and methods describing the functions each object can manage.

2.2.1 System Architecture

In any traffic simulator a mathematical model is needed to represent the transportation supply system simulating the technical and organizational aspects of the physical transportation supply. Second a demand model has to be generated to model the demand of persons and vehicles traveling on the supply system. Unlike macroscopic transport models, traffic control has to be modeled very detailed depending on supply and demand. Therefore the simulator contains three major building blocks plus one additional block generating the results of each simulation exercise.

The road and railway infrastructure including sign posts and parking facilities compose the first block. This block is needed to model the physical roads and tracks. Public transport stops and parking lots are needed as starting (origin) and ending (destination) points of trips. Since these are some physical and stationary network elements, they are also part of the first block. Finally fixed elements such as sign posts and detectors located on the road and railway infrastructure are also considered to be part of the infrastructure block.

The technical features of a vehicle and specifications of traffic flows are made in the second block. Traffic is either defined by origin–destination matrices or by generating traffic at link entries. The assignment model and path flow descriptions

are part of this block. Public transport lines are defined within this block as sequence of links and stops.

The traffic control block contains all elements required to control the traffic. Non-grade-separated intersections are controlled by rules to be defined in this block. Definitions for four-way stops, major/minor priority rules with gap acceptance, and control options of traffic signals are made within this block. Although a signal post with signal heads is belonging to the infrastructure block, the signalization itself including definitions about signal settings and actuated control belongs to the traffic control block.

All three blocks depend on each other indicated by arrows in Fig. 2.1. While running a traffic flow simulation vehicles (block 2) may activate detectors (block 1) which will influence vehicle-actuated signal control (block 3); thus during the simulation all three blocks are constantly activated with interdependencies between each block.

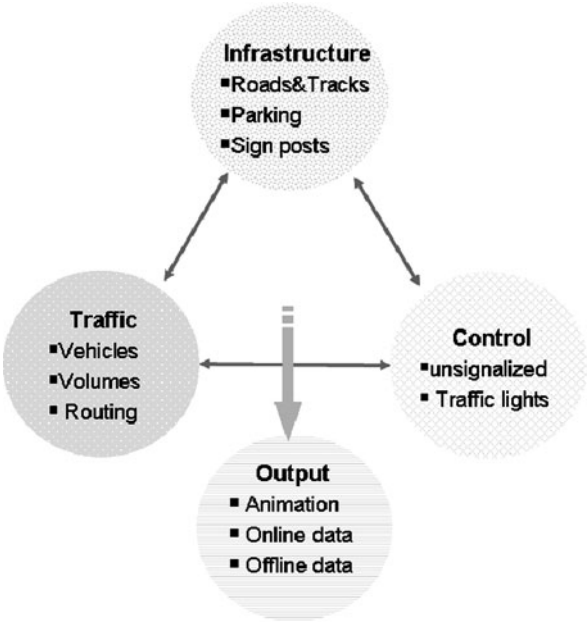


Fig. 2.1 Schematic representation of four building blocks

The situation is slightly different with the fourth block, which takes care of all kinds of data output. The evaluation block processes data provided by the first three blocks without a feedback loop. Output may be generated during the simulation either as animated vehicles and states of traffic control or as statistical data on detector calls and vehicle states presented in dialogue boxes. Most measures of performance (MOEs) are generated during the simulation, kept in storage and filed at the end of each simulation.

In the following sections, the classes of each block along with major attributes and features will be listed.

2.2.2 Infrastructure Modeling

The level of detail required for replicating the modeled roadway infrastructure depends on the purpose of a VISSIM application. While a rough outline of the analyzed intersection is sufficient for testing traffic-actuated signal logic, a more detailed model is required for simulation analyses. For the purpose of simulating traffic operations, it is necessary to replicate the modeled infrastructure network to scale. Scaled networks are imported from macroscopic transport planning and GIS software, signal optimization software, or manually traced based on scaled orthorectified aerial photographs and CAD drawings. The road and railway infrastructure is modeled by distinctive elements called classes; the most important ones are described in the following section.

2.2.2.1 Links and Connectors

Roadway networks are usually represented by graphs with nodes located at intersections and links placed on road segments. Nodes are needed if (a) two or more links merge, (b) links cross each other, (c) one link splits into two or more links, and (d) the characteristics of a road segment change. For the purpose of additional flexibility VISSIM does not require an explicit definition of nodes. However, the functionality of merging, crossing, and splitting is modeled by connectors to tie two links. Connectors will always connect pair wise; thus merging from one to three links will require three connectors.

Each link has certain mandatory and some optional properties describing the characteristic of the road or railway. The mandatory properties contain a unique identification, the planar coordinates of its alignment, number of lanes with lane widths, and the type of vehicles suited for the link. Optional properties are required for less standard simulation tasks such as z-coordinates for gradient sections, cost values for tolling sections, and particular driving behavior settings such as mixed traffic or banned lane usage for particular vehicles such as banned overtaking by trucks.

If the properties of a link do not change a link may continue for several road segments. In Fig. 2.2 an entry merges with a three-lane motorway. In order to allow continuous merging from the acceleration lane onto the motorway the three-lane link ends at the merging area and a four-lane link is tied to the previous motorway section and the on-ramp. Behind the merging an off-ramp occurs without an additional deceleration zone. Since the properties of the three-lane motorway do not change, the off-ramp may divert without splitting the modeled motorway at the point of diversion.

Connectors tie links based on lanes. Lane allocation and the visibility distance are important mandatory properties of a connector influencing the lane selection of vehicles. In order to fit the alignment of turning movements, the geographical shape of links and connectors can be aligned according to Bezier curves.

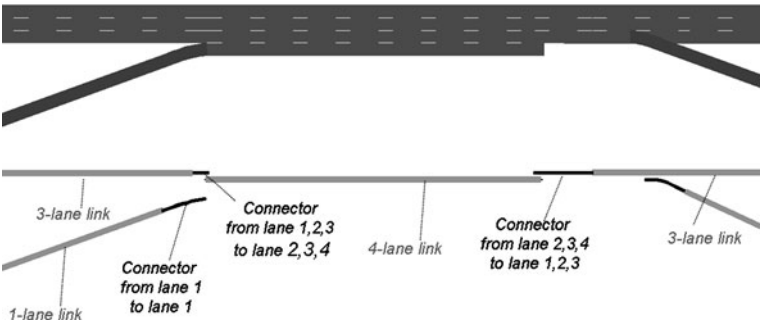


Fig. 2.2 Links and connectors modeling merging and diversion of roadways

2.2.2.2 Other Network Elements

Links and connectors are the basic building block needed to add other infrastructural objects. There are classes of spot objects and others with a spatial extension. The location of each object relates to a particular lane; thus objects relevant to full cross section must be copied for each lane. Spatial objects can only be extended on one link. A spot object does not have a physical length and has to be located at one particular point (coordinate) on a lane. Typical spot objects are as follows:

- Speed limit sign: As a vehicle passes this sign its desired speed will be adjusted according to the posted speed.
- Yield and stop signs: These signs identify the position vehicles of minor movements and will wait for major movements to pass.
- Signal head: A signal head is used to display Green and Red times. In the simulation the signal head is located at the physical stop bar. Vehicles will stop randomly distributed between 0.5 and 1.5 m before the signal stop.

Spatial objects start at a position on a lane and extend for a given length. The most prominent spatial objects describing the infrastructure are as follows:

- Detectors observe vehicles and people while passing a certain position. The message impulse may be used either for the purpose of statistical evaluation or for transmitting the data to the signal controller to be interpreted by the signal control logic. Detectors may have a length but a zero length is acceptable for push buttons. Detectors contain a type of discriminating pulse, presence, speed, and particular public transport detectors.
- Stop locations for public transport vehicles: Buses may stop either on the lane itself (curbside stop) or as lay-by turnout. Tram stops are always curbside stops. The length of a stop should be longer than the longest public transport vehicle; otherwise passengers may not board and alight from this vehicle.

- Parking lots are optional objects needed as origins and destinations if vehicles choose their routes by dynamic assignment. Victious parking lots are required for dynamic assignment but parking lots may also have a physical size if parking space availability will be looked at.
- Speed areas specify part of a link or connector with different desired speeds of vehicles. Typically speed areas are used for turning movements so that vehicles will turn with a reduced speed and accelerate thereafter to its previous speed.

2.2.3 Traffic Modeling

After the definition of the physical layout of a modeling system the vehicles traveling on the infrastructure have to be specified. While private transport vehicles may search for individual routes, public transport vehicles will follow predetermined routes with stops on their way. Buses on non-regular services such as sightseeing coaches should be modeled as private transport.

2.2.3.1 Private Transport

The class of private transport vehicles is classified in categories like trucks, cars, bikes, and pedestrians. Within each category a particular vehicle model with mandatory technical features like vehicle length, width, acceleration and deceleration rates, and maximum speed is defined. Depending on the purpose of the modeling application data entry of vehicles can be simplified by the specification of distributions of these technical features instead of defining individual vehicle types. The proper distribution of vehicle length reflecting the real vehicle fleet influences the simulation result such as queue length. For most studies vehicle width is irrelevant but modeling-mixed traffic requires the precise definition of the geometric extension of each vehicle type. The vehicle types can be aggregated to a set of vehicles for analysis purpose such as collecting the total travel time of all HOV vehicles. Thus a private transport vehicle is defined by the following attributes:

- Vehicle category-like modes (mandatory)
- Vehicle length or distribution of vehicle lengths (mandatory)
- Distributions of technical and desired acceleration and deceleration rates as a function of speed (mandatory)
- Maximum speed or distribution of maximum speeds (mandatory)
- Vehicle width (optional)
- Color and 3D model or distribution of colors and 3D models (optional)
- Vehicle weight or distribution of vehicles weights (optional)
- Emission class or distribution of set of emissions (optional)
- Variable and fixed cost of vehicle usage (optional)

Vehicles are generated randomly at link entries or at parking lots which may be located in the middle of link segments. Data input flows are defined individually

for multiple time periods. As the number of departures in a given time interval [0,t] follows the Poisson distribution with mean = λt, the time gap x between two successive vehicles will follow the exponential distribution with mean $1/\lambda$. λ is measured in vehicles per hour. The probability of a time gap x between two successively generated vehicles can be computed by

$$f(x, \lambda) = \lambda e^{-\lambda x} \tag{2.1}$$

If the defined traffic volume exceeds the link capacity the vehicles are stacked outside the network until space is available. It is noted if the stack is not emptied at the end of the simulation time.

2.2.3.2 Public Transport

All technical properties of private vehicles are also relevant for public transport vehicles but additional characteristics are added. A public transport line consists of buses, trams, or light rail vehicles serving a fixed sequence of public transport stops according to a timetable. The stop times are determined by the distribution of dwell times for boarding and alighting or calculations of passenger service times. The timetable describes the departure time at the initial stop. The departure times at the following stops is calculated by

$$\begin{aligned}&\text{Simulated arrival at next stop} + \text{dwell time} + \max(0; (\text{start time}\\ &\quad + \text{departure time offset} - \text{simulated arrival} + \text{dwell time})\ \text{Slack time fraction})\end{aligned} \tag{2.2}$$

The departure time offset is defined by the timetable between two successive stops. The slack time fraction accounts for early starts, if set below 1. If the scheduled departure time is later than the sum of arrival time and dwell time, the public transport vehicle will wait until the scheduled time is reached. If the slack time fraction or the departure time offset is 0, the vehicle will depart according to traffic conditions and dwell time only.

2.2.4 Traffic Control

2.2.4.1 Non-signalized Intersections

The right-of-way for non-signal-protected conflicting movements is modeled with priority rules. This applies to all situations where vehicles on different links or connectors should recognize each other. The priority rules are used to model the following:

- Uncontrolled intersections where traffic has to give way to traffic on the right
- Uncontrolled intersections where traffic on the terminating road must give way to traffic on the continuing road

- Two-way stop-controlled and all-way stop-controlled intersections
- Roundabouts where vehicles entering the roundabouts have to give way to traffic within the roundabout
- Merging zones where traffic entering from a ramp has to yield on traffic of the major road
- Semi-compatible movements (permitted turns) at signalized intersections such as right and left turns conflicting with parallel pedestrian movements or left turns parallel to opposing through movements
- Buses leaving a lay-by stop have right-of-way if they indicate their movement

Due to the flexibility of the priority rules different national guidelines can be considered. For simplicity the following notes refer to right-hand traffic and may be shifted, if applied to left-hand traffic. VISSIM provides an option that the necessary conversions are made automatically.

The priority rule consists of a stop line indicating a waiting position for vehicles of minor movements (vehicle 2 in Fig. 2.3). At the stop line the minor vehicle will check if a vehicle of the major movement (vehicle 1) is within the headway area. The headway area is defined as a segment starting slightly before both movements merge. The position itself will be set manually. Additionally the minor vehicle checks if a major vehicle will reach the conflict marker within the minimum gap time if traveling with its present speed. Vehicles will only stop at intersections with a yield sign if a major vehicle is either in the headway area or within the gap time zone.

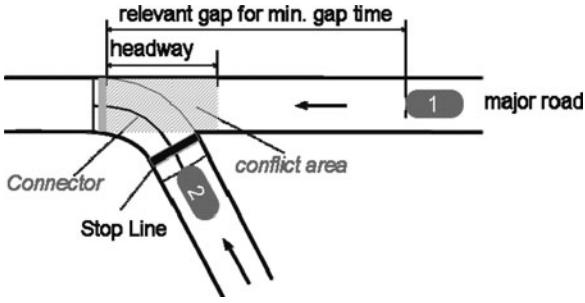


Fig. 2.3 Concept of modeling priority rules

2.2.4.2 Signalized Intersections

While the signal heads are part of the infrastructure the signal displays belong to the traffic control block. Multiple signal heads with identical signal display belong to one signal group. A signal group is the smallest entity to be controlled by a signal control unit. As such the two lanes of the westbound straight through movements in Fig. 2.4 are signaled by group K3 while the right and left turn may have different signal settings. After allocating signal heads to signal groups, clearance times between conflicting movements are defined within an intergreen matrix. If the volumes of each movement are already defined by routes, an optimization routine is available to calculate delay minimizing signal settings for a given cycle time.

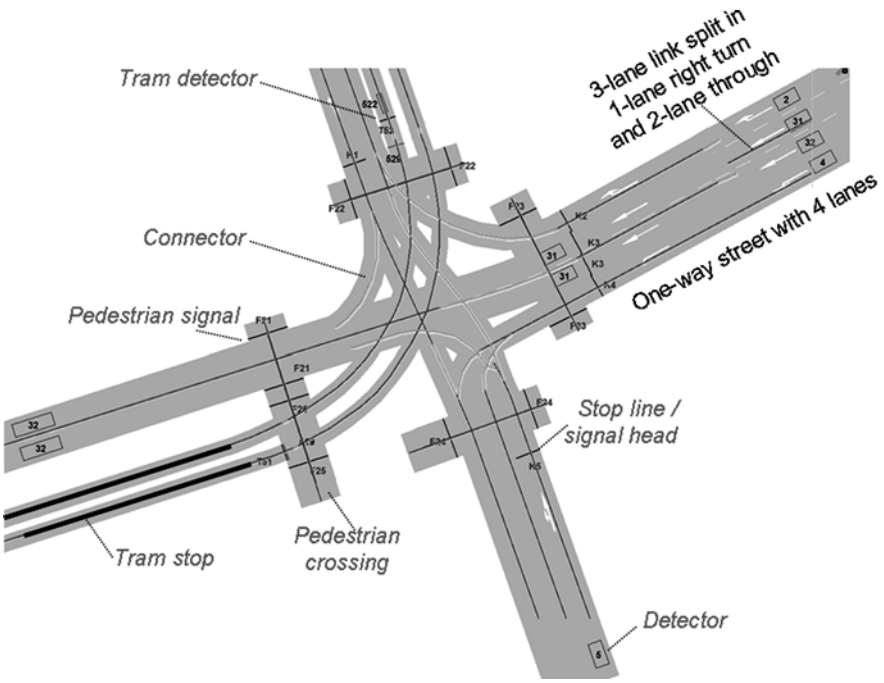


Fig. 2.4 Urban signalized intersection with tram tracks

In the Highway Capacity Manual and other national guidelines numerous analytic formulae are available to estimate delay, queue length, and number of stops for fixed-time signal settings (pretimed control). Therefore microsimulation is rather used for actuated and adaptive signal control to account for the stochastic influence of vehicle arrival and the feedback loop between vehicle arrivals and signal settings. VISSIM contains a programming language with a graphical flowcharter to define actuated signal control. It is a structured programming language like C or Pascal added with some functions relevant for traffic engineers. For example, functions are available to receive detector pulses, occupancy rates, and presence. Furthermore the display of signal groups and stages can be accessed. An actuated logic can be defined based on signal groups or based on stages and interstages to reflect national standards of signalization. Since signal control is handled very different in North America an external interface is available for ring-barrier control (Fig. 2.14).

2.2.5 Data Output

Vehicle movements may be animated in 2D or in 3D. This feature allows users to create realistic video clips in AVI format, which can be used communicating a project's vision. For better representation background mapping capabilities with aerial photographs and CAD drawings should be applied. Additionally building

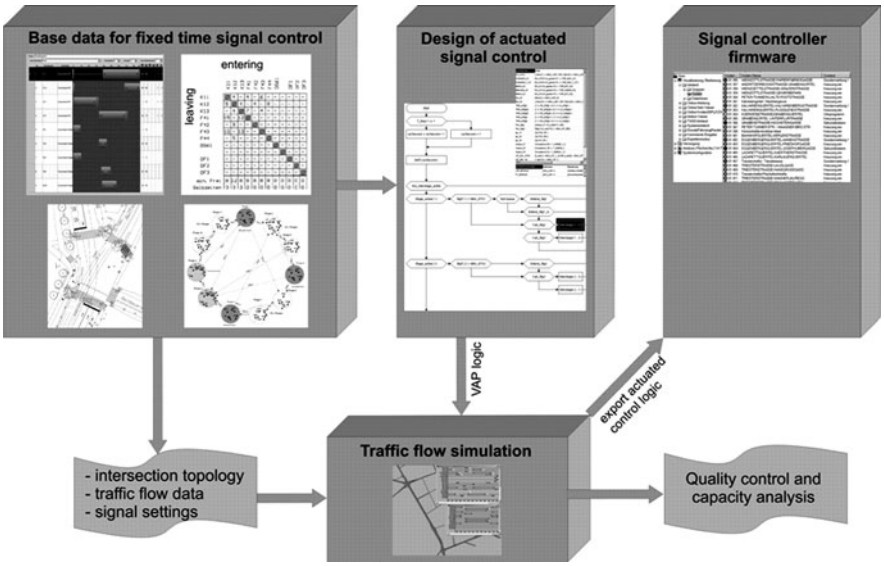


Fig. 2.5 Modeling actuated signal control using vehicle-actuated programming (VAP)

models can be imported from Google Sketchup®. For even more advanced virtual reality visualization, the simulated traffic can be exported to Autodesk® 3DS Max software.

Numerous measures of effectiveness (MOEs) are being reported. Typical MOEs include delay, travel time, stops, queues, speeds, and density. The decision-making process is supported by providing the flexibility to summarize and report the MOEs needed to answer the problem. When, where, and how data are reported from a simulation is defined by the user. Data can be summarized for any time period and interval within that time period; at any point location in the network, at an intersection, along any path, or for the entire network; and aggregated by any combination of mode, or individual vehicle class. Data can also be reported for an individual vehicle. Data are provided in ASCII or database formats and automatically formatted using common software such as Microsoft Access or Excel. Several MOEs can also be exported to the transportation planning software, VISUM, for detailed graphical representations. VISUM provides an extensive graphics library for effectively visualizing transportation modeling results (Fig. 2.6).

2.3 Fundamental Core Models

2.3.1 Car Following

Michaelis (1963) introduced a concept that a driver will recognize changes in the apparent size of a leading vehicle as he approaches this vehicle of lower speed. Speed differences are perceived through changes on the visual angle. Minimum

Fig. 2.6 3D representation of a simulated urban intersection including trams and cyclists

changes over time are required to be recognized by drivers. Experiments indicate certain thresholds on relative speed difference and distance for drivers of the lagging vehicle to take an action. These types of car-following models are considered to be psycho-physical car-following models, also referred to as action point models.

Approaching a slower leading vehicle some action points are recognized by the driver while others are being done unconsciously. The action point of conscious reaction depends on the speed difference, relative distance to the leading vehicle, and driver-depended behavior. There are four different stages of following a lead vehicle. Figure 2.7 indicates the oscillating process of this approach. The thresholds are explained in an abbreviated form. Driver-specific abilities to recognize speed differences and individual risk behavior are modeled by adding random values to each of the parameters as shown for *ax*. For a complete listing of the random values the reader is referred to Wiedemann and Reiter (1992).

ax: Desired distance between the fronts of two successive vehicles in a standing queue. $ax := VehL + MinGap + rnd1 \cdot axmult$ with *rnd1* normally distributed N(0.5, 0.15).

abx: Desired minimum following distance which is a function of *ax*, a safety delta distance *bx*, and the speed with $abx := ax + bx \cdot v$.

sdv: Action point where a driver consciously observes that he approaches a slower leading car. s*dv* increases with increasing speed differences Δv. In the original work of Wiedemann (1974) an additional threshold *cldv* (closing delta velocity) is applied to model additional deceleration by usage of the brakes with a larger variation than *sdv*.

opdv: Action point where the following driver notices that he is slower than the leading vehicle and starts to accelerate again. The variation of *opdv* is large compared to *cldv*.

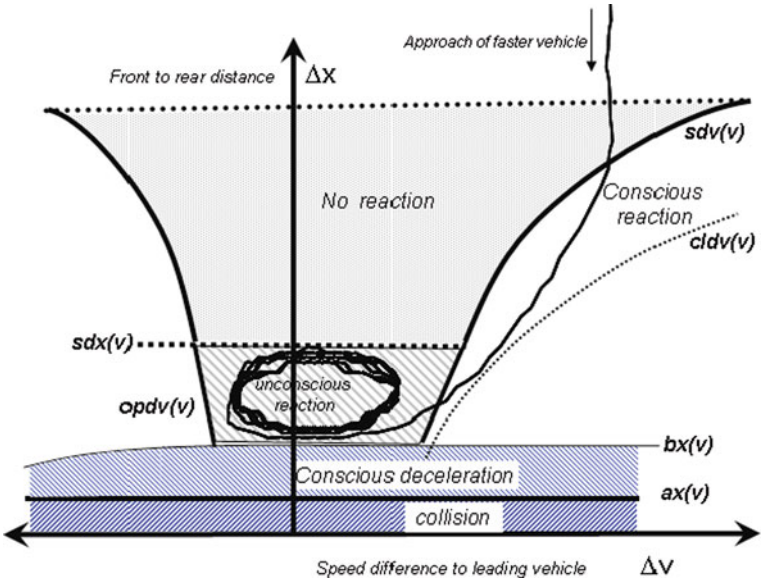


Fig. 2.7 Psycho-physical car-following model by Wiedemann

sdx: Perception threshold to model the maximum following distance which is about 1.5–2.5 times *abx*.

A following driver reacts to a leading vehicle up to a certain distance which is about 150 m. The minimum acceleration and deceleration rate is set to be 0.2 m/s^2. Maximum rates of acceleration depend on technical features of vehicles which are usually lower for trucks than the personal desire of its driver. The model includes a rule for exceeding the maximum deceleration rate in case of emergency. This happens if *abx* is exceeded. The values of the thresholds depend on the present speed of the vehicle as indicated by (*v*) for all thresholds in Fig. 2.7.

In Fig. 2.8 the impact of different threshold values is presented. Risk averse and less alert drivers do not follow as close as more risky drivers. The alertness is measured by *sdv*, the safety distance by *bx*. According to the fundamental diagram in the figure the road capacity of a two-lane motorway will account for about 1950 veh/h/lane, if all driver-vehicle units will drive risk aversely. Road capacity is increased up to 2250 veh/h/lane with thresholds set for more risky drivers. Different threshold values may lower or increase capacity accordingly.

2.3.2 Lateral Movements

Lateral movement in VISSIM can be structured in lane selection, lane-changing, and continuous lateral movement within one lane.

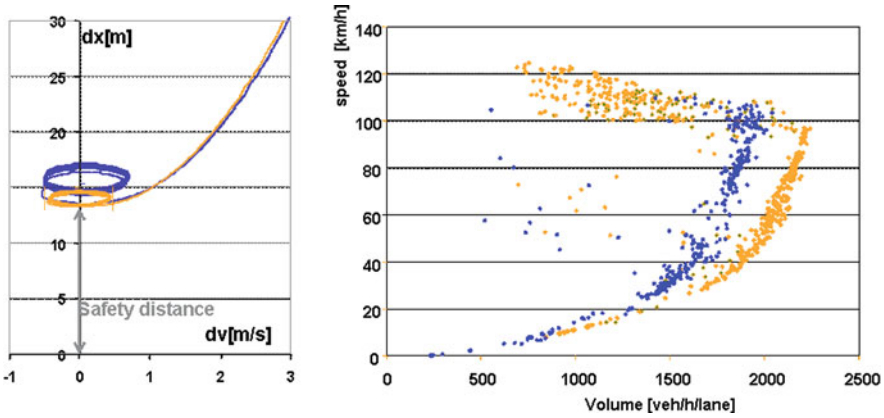


Fig. 2.8 Risk averse (*blue*) and more alert (*orange*) vehicle influences road capacity

2.3.2.1 Lane Selection

As long as a driver is not aware of any necessary lane change because he is far away from the next relevant intersection, he chooses the lane with the best interaction situation. Three tests are performed: First the driver decides if he wants to leave the current lane. This is the case whenever the interaction state (i.e., the driving mode in Wiedemann's car-following model) is different from free. Then he checks the neighboring lanes if there is a better interaction situation, i.e., either free or a higher time-to-collision. If one of the neighboring lanes provides a better situation downstream, the last check is if a lane change is possible considering the vehicles upstream, what is modeled as gap acceptance described below in more detail.

However, lane selection is often governed by mandatory lane changes for desired turns at junctions downstream. In VISSIM's network coding, each connector has two distances attached: the lane change distance and the emergency stop distance. The lane change distance describes when a driver becomes aware of the upcoming connector; typical values range between 100 and 500 m. From that point on he will consider the connector in his lane selection. The emergency distance is the distance to the connector where a driver will stop when he was not able to reach the necessary lanes to change to the connector. A connector leaving a link is typically attached to only some of the link's lanes, e.g., a single-lane right turn on a three-lane main road link will be connected only to the rightmost lane of the link. This means a car must be on the rightmost lane to change to the connector.

A driver who follows a defined route knows which connectors he has to take to follow this route. All these connectors have lane change distances, and there might be parts of the route where the driver is in the lane change distance of several connectors at the same time (when the distance between the connectors is less than the lane change distances). The driver takes into account all connectors that he is aware of when selecting a lane. The desired lane is the one that allows him to follow

his route through the upcoming connectors with the least number of mandatory lane changes.

In the following example the lane selection is visualized using a turbo roundabout with spiral markings. Drivers intending to make 270° turns should approach at the left lane. This is modeled by lane-changing distances which will extend prior to the approach lanes as shown for the west and south approach in Fig. 2.9 by the amber shading. Connectors are presented in purple and links in light blue.

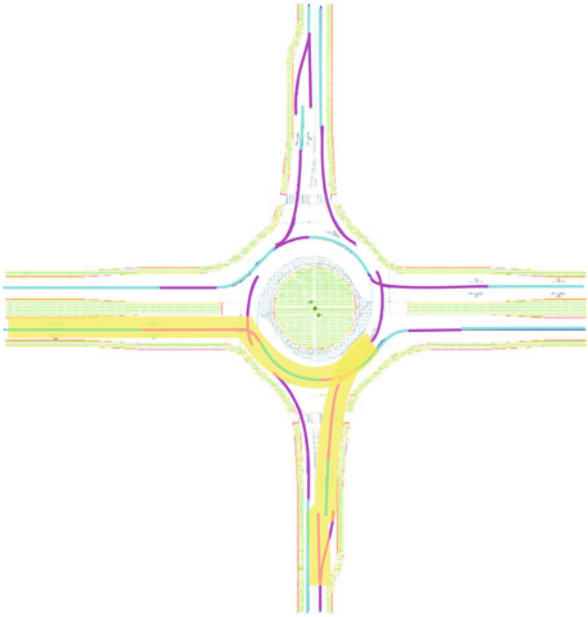


Fig. 2.9 Visibility of lane-changing distance at a multi-lane roundabout with spiral marking

2.3.2.2 Lane Changing

The actual lane-changing logic in VISSIM is used to decide if it is possible to change to the desired neighbor lane or not. The desired lane is a result of the lane selection process for either free or mandatory lane changes based on gap acceptance: A driver is willing to accept that he forces a lag vehicle on the desired lane to decelerate. The value of this accepted deceleration is a matter of calibration, and for mandatory lane changes it is as well a function of the distance to the emergency stop position of the next connector where the lane change has to be completed, i.e., the driver becomes more aggressive closer to the point of an emergency stop. In a similar way, the driver is willing to accept to decelerate himself in case of a mandatory lane change.

The accepted deceleration values for lag vehicles upstream and for the vehicle itself are parameters of the behavior model and can be defined selectively for pairs of link and vehicle types.

2.3.2.3 Continuous Lateral Movement

The lane-oriented driver behavior described above is complemented by space-oriented movement behavior where the lateral movement is not limited to instantaneous lane-changing maneuvers. Continuous lateral movement is allowed within and between lanes, including overtaking within the same lane sufficient lane width provided. Heterogeneous traffic conditions as found in many developing countries require modeling a mix of different vehicle types and the lack of lane discipline. Lane discipline helps to simplify the behavior models in simulation tools because it allows to structure the behavior in longitudinal behavior handled by the car-following logic and in lateral behavior reduced to lane changing, i.e., the lateral movement is discretized in steps of whole lanes. Typical heterogeneous traffic situations require continuous lateral movements only restricted by the road width. Furthermore the traditional gap acceptance models for lane changing are not suitable since the definition of a gap on the neighboring lane fails in the more space-oriented movement of the heterogeneous vehicles. And finally, the decision for lateral and longitudinal acceleration cannot be considered as independent processes.

Lane width is an optional attribute of links and connectors. By default vehicles are shaped rectangular, but diamond-shaped forms for bicycles allow for a more realistic queuing of two-wheelers at intersections. The driver behavior for lateral movement has to handle two situations not included in traditional car-following and lane-changing models: The drivers must choose a lateral position within the lane and the longitudinal behavior must incorporate more than one leading vehicle on the same lane.

The choice of the lateral position in a lane follows a simple but effective principle: The driver chooses the lateral position where he has the maximum longitudinal time-to-collision. To find this position, the driver divides the available road width in virtual lanes. These virtual lanes are constructed from the right and left sides of the preceding vehicles on the road, including some lateral safety distance. The lateral safety distance that a driver wants to keep while passing another vehicle depends on both vehicle types and on the speed of the overtaking vehicle, i.e., on the maximum of both speeds. A linear relationship of the safety distance from the speed is assumed, the user can define a minimum safety distance at very low speeds and a safety distance at 50 km/h.

To move to the desired position from his current position the driver applies a lateral speed depending on vehicle type and on his longitudinal speed. Under free driving conditions, a preferred lateral position is assigned to each driver being center of the lane, or on the left or on the right, or random. Thus it is possible to model that cars tend to obey to lanes in free traffic but break lane discipline in intersection areas or under congested traffic conditions.

The car-following model takes into account lateral safety distances when deciding which of the preceding vehicles are relevant for the acceleration behavior. Based on the current lateral position, the longitudinal model looks at all vehicles ahead within a user-defined look-ahead distance and determines which vehicles are relevant as leading vehicles. These are all vehicles which the driver cannot pass at the

moment with its current speed and lateral position, taking into account the lateral safety distances. Then for each of the relevant vehicles an interaction according to the psycho-physical car-following model is computed. The final longitudinal acceleration applied is the minimum of all accelerations that resulted from the individual interactions.

This lateral behavior model yields a good qualitative representation of the traffic conditions found, for example, on Indian roads. The quantitative calibration of the model is possible as well. Hossain (2004) calibrated the heterogeneous traffic model to match saturation flows measured by video at an intersection in the city of Dhaka, Bangladesh. Other examples of calibration work on this model is the application to bicycle traffic in Germany described in Falkenberg and Vortisch (2003) and on motorcycles in Vietnam. The latter work was presented by Matsuhashi et al. (2005). He measured speed–flow graphs of motorcycles using video analysis and calibrated the lateral safety distances to get a good macroscopic fit of the simulated speed–flow relationships.

2.3.3 Tactical Driving Behavior

Tactical driving behavior is typically defined as the driving behavior that requires some kind of planning ahead with a temporal horizon of more than one time step or a spatial horizon of more than just the direct neighboring vehicles. In order to build a simulation tool usable for practical planning work, the pure, academic car-following and lane-changing models need to be extended for a lot of special situations. In the following subsections, two extensions are described which are clearly in the domain of tactical behavior.

2.3.3.1 Anticipated Driving at Conflict Areas

A conflict area exists in VISSIM by definition wherever two links overlap. Right-of-way conditions can be defined. Individual priority rules are either defined by the user as described above or conflict areas are used where drivers compute a plan to cross the conflict area. A yielding driver observes the approaching vehicles in the main stream and decides when to go. Then he plans an acceleration profile for the next seconds that will allow him to cross the area, taking into account the situation behind the conflict area. If the driver realizes that he has to stop or to slow down due to other vehicles, he will calculate more time to cross the conflict area or decide not to go at all. He anticipates the behavior of the vehicles behind the conflict area, estimating if a car will accelerate or decelerate.

Vehicles in the main stream react to the conflict area as well: If a crossing vehicle could not complete the crossing because of the driver's overestimation, the vehicle in the main stream will slow down or even stop. If a queue builds up from a signal downstream of the conflict area, the vehicles in the main stream try not to stop on the conflict area in order not to block the crossing stream. This is accomplished

by having the drivers make a similar plan to cross the conflict area as the yielding vehicles do.

The extension of the behavior model was generalized in VISSIM so that drivers always have a plan of their acceleration profile for the next seconds which can be accessed by other drivers to model their anticipation of the behavior of observed cars. Of course the planned acceleration profiles cannot always be realized due to unexpected situations.

2.3.3.2 Cooperative Merging

In the chapter about lateral movement the process of selecting and changing a lane in the course of a mandatory lane change was already described. Even the increased aggressiveness toward the emergency position of a connector is not sufficient to model merging situations in dense traffic. To achieve realistic merging capacities it is necessary to include two aspects in the model: an influence of the lane change situation on the longitudinal behavior of the merging car and cooperation of the drivers in the main stream. Obviously this cannot be part of a car-following model alone, because it requires interaction between longitudinal and lateral behavior.

To consider the first aspect a merging driver (or actually any driver that is in a mandatory lane change situation) observes the gap situation on the desired lane. If there is no gap immediately available, he adapts his speed to the average speed level of the desired lane in order to reduce the necessary decelerations after a lane change. If he comes closer to the point where the lane change must be completed, he starts slowing down to reach the next upstream gap in the desired lane.

The second aspect, cooperation, is implemented by giving drivers information about vehicles in a mandatory lane change situation aiming to the lane they are currently using. If a driver sees a merging vehicle on the neighbor lane that could possibly change into his lane in front of him, he applies a user-defined deceleration to keep the gap open or even widen it. The lane-changing vehicle is made aware of the cooperating driver and can take that into account in his lane change decision. In order not to block the mainstream by too much cooperation with a heavy merging stream, a driver will cooperate only once within a definable distance.

In case of congested weaving another situation is treated especially. If speeds are very low and two vehicles want to swap lanes but are driving side by side, then one of the vehicles will slow down to open a gap avoiding a mutual blockage.

2.3.4 Pedestrian Modeling

For cars, motorcycles, and bicycles the lateral movement model described above is appropriate, because these vehicles all have still a dominant longitudinal movement. For pedestrians this is not as clear, since pedestrians can move sideways even without moving longitudinally. Therefore, another movement model is required for pedestrians allowing a more area-based behavior compared to the non-lane-based behavior of vehicles.

The movement of pedestrians is based on the Social Force Model (Helbing and Molnár, 1995). The basic idea is to model the elementary impetus for motion with forces analogously to Newtonian mechanics. From the social, psychological, and physical forces a total force results, which then sums up to the entirely physical parameter acceleration. The forces which influence a pedestrian's motion are caused by his intention to reach his destination as well as by other pedestrians and obstacles. Thereby other pedestrians can have both an attractive and a repulsive influence.

The driving force to the destination is deduced from a so-called floor field, i.e., is a function of the position (x, y) of the pedestrian that gives the direction of the shortest path to the destination. The repulsive forces from walls and other pedestrians can be modeled in various ways. Several variants of the original social force model have been developed.

In order to incorporate the social force model including recent extensions, the data model of VISSIM and the user interface had to be extended in order to handle the areas on which pedestrians move. Areas can have different types and be associated with different parameter sets for the behavior model. The floor field is implemented in form of a grid with user-defined grid size, typically 10 cm.

An interesting aspect is the interaction of pedestrians and vehicles in the simulation. The interaction can be modeled at intersections in the form of signal-controlled or priority-controlled conflict areas. Pedestrians may not obey the signals but instead look for gaps in the vehicle stream.

The original social force model reproduces the movement of large pedestrian crowds very well. In 2006 it was used to model the movement of pilgrim crowds for the re-planning of the infrastructure in Mecca. For the use in traffic engineering applications, the version implemented in VISSIM had to be calibrated mainly to reproduce capacities, i.e., the maximum possible flow rates at bottlenecks. The necessary field data needed for the calibration can be collected from video observations of real-world situations or from controlled experiments. A detailed explanation of these calibrations is given in (Kretz et al., 2008). Additionally, it was assured that microscopic effects like lane formation in counterflow situations and stripe formation in crossing flow situations are reproduced and that a realistic impression of resulting animations was generated (Fig. 2.10).

2.3.4.1 Path Choice

For any traffic simulation system it is essential to model realistically the volume of generated vehicles and the routes these vehicles will take. The input volumes are described above. This section will present three different ways to define routing.

2.3.5 Fixed Routes

A route is a fixed sequence of links and connectors: A route may have any length – from a turning movement at a single junction to a route that stretches throughout an entire simulation network. A route starts at a routing decision and ends at destination

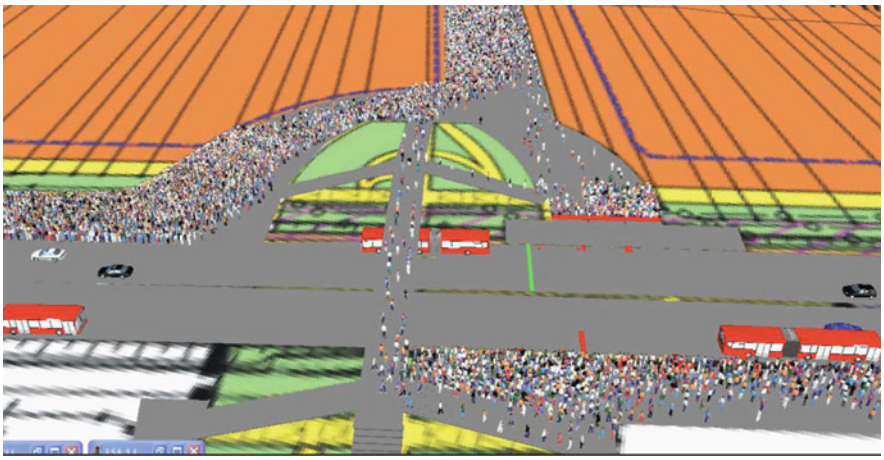


Fig. 2.10 Pedestrian bridge between stadium and bus stop 50 min after the finish of a soccer game

point. Each routing decision point may have multiple destinations resembling a tree with multiple branches. A routing decision affects all or a subset of vehicles types. Vehicles usually travel on one route before entering a new route. However, partial routes are available allowing to distribute vehicles on alternative subsets of routes. Managed lanes for high occupancy traffic are typically modeled with partial routes using a user-definable toll pricing model.

If vehicles travel on fixed or partial routes they are aware of downstream turning movements if the visibility distance (lane selection distance) reaches upstream to the particular point (see Fig. 2.9); thus the combination of visibility distance and predefined routes is a crucial element to model lane selection realistically. Both routing decisions and lane selection distance can be defined for all vehicles or subsets of vehicles. Quite often the set of driver-vehicle units is subdivided in a class of drivers familiar with the area and another class following street signs. This feature is widely used to model complex situations where some drivers tend to select appropriate lanes several intersections ahead, while others are decide and maneuver late.

2.3.6 Dynamic Routing

Predefined routing is not appropriate for applications which require a feedback between traffic volumes and the decision of preferred routes. Any route guidance system considers past, current, or anticipated volumes in order to regulate and manage traffic movements. The programming language VAP is not only used for signal control but can also be applied for setting routes dynamically. Depending on detector values, vehicles movements can be rerouted. A small logic has to be designed describing traffic conditions and rerouting decisions. The rule-based

feature of dynamic routing with VAP is applied for routings in small areas (e.g., booth selection at toll plazas) as well as large networks such as route guidance on alternative motorways. Like fixed routes, dynamic rerouting can also be applied on a subset of vehicles.

2.3.7 Dynamic Assignment

General Approach

Traffic assignment is essentially a model of the route choice of the drivers or transport users in general. For such a model it is necessary first to find a set of possible routes to choose from, then to assess the alternatives in some way, and finally to describe how drivers decide based on that assessment. The modeling of this decision is a special case of discrete choice modeling, and a lot of theory behind traffic assignment models originates from the discrete choice theory.

The motivation to include route choice in a simulation model like VISSIM is twofold: With growing network size it becomes impossible to supply the routes from all origins to all destinations manually, even if no alternatives are considered. On the other hand the simulation of the actual route choice behavior is of interest because the impacts of control measures or changes in the road network on route choice are to be assessed.

The dynamic assignment procedure in VISSIM is based on the idea of iterated simulation. That means a modeled network is simulated not only once but repetitively and the drivers choose their routes through the network based on the travel cost they have experienced during the preceding simulations. Formally speaking the process aims at computing dynamic stochastic user equilibrium.

Network Coding for Dynamic Assignment

Assignment-related problems typically refer to a more abstract idea of the road network than a typical detailed network topology with links and connectors suited for microscopic simulation. Usually intersections are nodes and roads between the intersections are edges of an abstract graph. The assignment procedures can operate much more efficiently on this type of graph, and this level of abstraction is more appropriate even for the human understanding of the problem. Therefore an abstract network representation is built for the dynamic assignment and the user defines the parts of the network model that are to be considered as nodes. Normally real-world intersections are defined as abstract nodes. The sequence of links and connectors between two adjacent nodes is defined as an edge. The dynamic assignment is calculated based on the node–edge topology.

Travel demand for dynamic assignment is specified by an origin–destination matrix. To define travel demand using a origin–destination matrix, the area to be simulated is divided in sub-areas called zone**s** and the matrix contains the number of trips that are made from all zones to all zones for a given time interval. To model the points where the vehicles actually appears or leaves the road network, a network

element *parking lot* is used. A parking lot belongs to a certain zone, i.e., trips originating from this zone or ending in this zone can start or end at this parking lot. A zone can have more than one parking lot. The total originating traffic of a zone is distributed to its parking lots according to relative flows. One parking lot can belong to one zone only.

Alternatively it is also possible to supply traffic demand as a set of trip chains. A trip chain describes for a vehicle a series of zones it will drive to and the time it will spend there. In contrast to O–D matrices a trip chain file allows to supply the simulation with more detailed travel plans for individual vehicles; however, the coding effort is much higher.

Route Search, Assessment, and Choice

During a simulation, travel times are measured for each edge in the abstract assignment network. All vehicles that leave the edge report the time they have spent on the edge. All travel times during one evaluation interval are averaged and thus form the resulting travel time for that edge. There is a special treatment of vehicles that spend more than one evaluation period on an edge, e.g., during congestion. They report their dwell time as well, although they have not left the edge. That is necessary to get information about heavily congested links even if there is no vehicle able to leave because of congestion.

Travel times per edge measured during a time interval of one iteration are exponentially smoothed before they are used in the route choice decision:

$$T_a^{n,\kappa} = (1-\alpha) \cdot T_a^{n-1,\kappa} + \alpha \cdot TO_a^{n,\kappa} \tag{2.3}$$

where

K = index of the evaluation interval within the simulation period
n = index of the assignment iteration
a = index of the edge
$TO_a^{n,\kappa}$ = measured travel time on edge a for period k in iteration n
$T_a^{n,\kappa}$ = expected travel time on edge a for period k in iteration n
α = smoothing factor

The route choice decision is based on a general cost function which is a linear combination of travel time, travel distance, and financial cost (e.g., tolls). The weights of the cost components can be defined by the user separately for user-defined vehicle classes. A route is a sequence of edges that describes a path through the network. Routes start and end at parking lots. The general cost for a route is defined as the sum of the general costs of all its edges.

Obviously, one would like to know the set of the n best routes for each origin–destination pair. Unfortunately there is no efficient algorithm to simply compute the n best routes but there are algorithms to find the single best one. To solve this problem a search for the best route for each O–D pair is activated in each iteration of the dynamic assignment. Since the traffic situation and thus travel times change from iteration to iteration (as long as convergence is not reached) different "best" routes will be found in the iterations. All routes found (i.e., all routes that have

qualified at least once as a best route) are collected in an archive of routes and are known in all later iterations.

Since in the first iteration no travel time information from preceding simulation runs is available, the cost is evaluated by replacing the travel time with the distance. Thus for the initial route search also link/connector costs are taken into account. For every subsequent iteration the edges in the network that have not been traveled by any vehicle have a default travel time of only 0.1 s, so that it attracts the route search to build routes including unused edges. This may lead to useless routes in the route collection. A route is considered useless if it is an obvious detour, and an obvious detour is defined as a route that can be generated out of an other known route by replacing a sequence of links by a much longer sequence (in terms of distance). How much longer the replacing link sequence must be to qualify as a detour can be defined by the user.

For the distribution of the travel demand of an O–D pair to the known paths, Kirchhoff distribution formula is used:

$$p(R_j) = \frac{U_j^k}{\sum_i U_i^k} \tag{2.4}$$

where

U_j = utility of route j
$p(R_j)$ = probability of route j to be chosen
k = sensitivity of the model

Actually the Kirchhoff distribution formula can be expressed as a Logit function, if the utility function is transformed to be logarithmic:

$$p(R_j) = \frac{U_j^k}{\sum_i U_i^k} = \frac{e^{k \cdot \log U_j}}{\sum_i e^{k \cdot \log U_i}} = \frac{e^{-k \cdot \log C_j}}{\sum_i e^{-k \log C_i}} \tag{2.5}$$

where C_j is the general cost of route j.

VISSIM offers an optional extension of the route choice model to correct the biased distribution in case of overlapping routes. It is based on the idea of the commonality factor of the routes (Cascetta et al., 1996) and implements a special case of the C-Logit model, adapted to VISSIM's Kirchhoff formulation of the route split.

2.4 Calibration and Validation

In the 15 years of VISSIM's existence lot of calibration efforts have been undertaken to adjust the parameters of the behavior models to the observed driving behavior in different countries in the world. Since microscopic trajectory data is difficult to get,

most of these efforts used macroscopic data provided by standard cross-sectional measurement, typically being volume and speed information in short time intervals.

One of the most recent of the more comprehensive efforts to calibrate VISSIM to macroscopic data is reported in (Menneni et al., 2008). In this study calibration was not based on time profiles of speed and volume but on scatter plots of speed–flow-pairs. These diagrams are very useful for calibration because they contain information about a broad range of traffic situations. In particular they show how the traffic flow behaves around capacity. This is the reason why these speed–flow diagrams are often used for comparing simulation results with real-world data.

The calibration procedure applied was using a genetic algorithm to systematically modify the parameters of the behavior model in order to fit the speed–flow diagrams from measurements and simulation. The measurement data came from a section of US Freeway 101 and were provided as a result of the NGSIM research program in the United States (Kovali et al., 2007). For the application of the genetic algorithm, a distance measure for speed–flow diagrams was developed that served as a measure of fitness in the algorithm (Fig. 2.11).

The calibration study generated parameter sets that reproduced the real speed–flow diagrams very well. An interesting aspect in this work is that the genetic algorithm provided for some parameter results that were very close to the default parameters in VISSIM, although the start values for the algorithm were quite different. For some other parameters the procedure suggested values different from the defaults provided at the time of the study. The study was therefore extended to look at microscopic trajectory data (Menneni et al., 2009) which were available for the same freeway section from the NGSIM project. As a result, some of the parameters for the driving mode following in Wiedemann's model could be adjusted to better reflect American driving style on freeways.

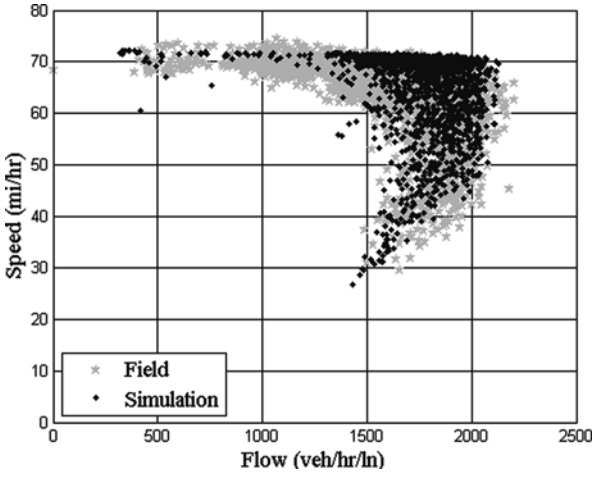


Fig. 2.11 Speed–flow based on calibration results for US 101 NB

2.4.1 Calibration Based on Microscopic Data

Lately Hirschmann and Fellendorf (2009) conducted some calibration based on single-vehicle trajectories. A vehicle fleet was equipped with high-precision GPS. The vehicles were traveling 3 days using a predefined route on an urban signalized arterial with adaptive signal control. Signalization varied heavily on 3 out of the 11 signalized intersections due to extensive bus and tram priority. The vehicle position was recorded at a rate of 20 Hz and was supplemented by travel time measurements based on automatic number plate recognition. Calibration on a microscopic level allows evaluating parameters of single-vehicle movements such as accelerations at each time step. Figure 2.12 shows measured and modeled acceleration rates over speed. It was anticipated that acceleration rates decrease as speed increases. However, the measurements indicate that the maximum acceleration rate can be observed at speeds around 10 km/h within a city center environment. It was obvious that the desired acceleration rates were too high by default and the mean acceleration rate has been lowered for this study. Using the modified acceleration rates the mean simulated trajectories fit well with the mean measured trajectory. Maximum speeds, the duration of maximum speeds and acceleration–deceleration rates compare well, although the simulated trajectory has a rather late start. The mean shown in the Fig. 2.12 was taken using 30 measured driving cycles and selecting 30 simulated trips randomly. It is noteworthy that the vehicles are hardly traveling with constant speed.

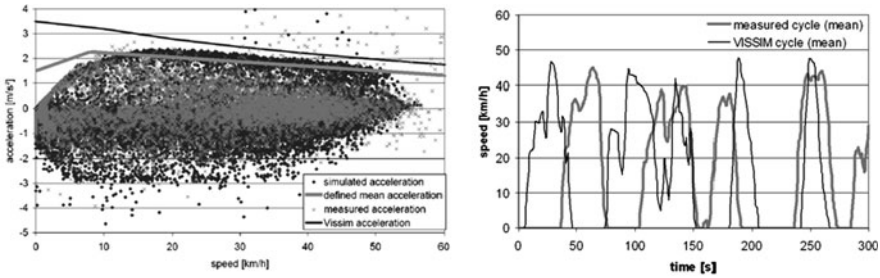


Fig. 2.12 Microscopic calibration using acceleration–deceleration rates over speed (*left*) and mean trajectory data measured with GPS (*right*)

2.5 Interfaces to External Applications

2.5.1 Application Programming Interface

VISSIM implements Microsofts Component Object Model as a programming interface. The functionality provided by a COM interface can be used by different programming languages; among them popular scripting languages like Visual Basic or Python. If any functionality is provided by a COM interface, all the objects dealt

with (like the vehicles in VISSIM, for example) and the operations on these objects must be modeled in the COM object model. Within the object model, the objects are structured in meaningful groups and hierarchies. So links may be found enclosed in a superior object called *net*. The links themselves contain the lanes and so on. The COM interface provides access to

- the modeled road network with all its attributes
- signal control
- evaluations
- all vehicles in the simulation and their attributes
- the simulation control, i.e., the simulation, can be started and stopped, and parameters can be read and set (Fig. 2.13).

The COM interface can be used to include VISSIM in other applications, e.g., if a user wants to write his own user interface to a set of special simulations, or to influence the behavior within the simulation. The COM program can run the simulation time step per time step and modify the state of vehicles or any network objects between the steps. For some decision models in VISSIM (e.g., route split, or the pricing model for HOT lanes) callback functions are provided at the COM interface so that the user can supply his own functions to override VISSIM's built-in models.

The COM interface is also widely used to simplify and document repetitious simulation tasks. Project automation using scripts has also proven to reduce errors for complex simulation task as repetitive tasks are recorded and used later for different scenarios to be modeled.

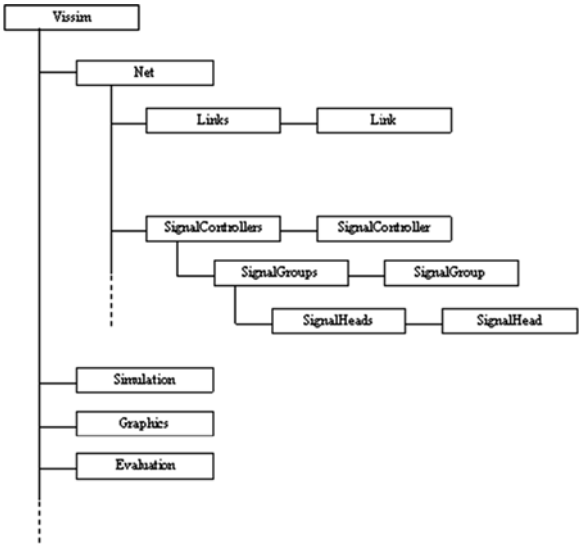


Fig. 2.13 Object hierarchy of VISSIM's COM interface

2.5.2 External Signal Controllers

Testing and analyzing signal control is one of the primary objectives of VISSIM. Therefore a flexible interface is provided for external signal controllers.

There are several ways to model signal control in VISSIM:

- Fixed-time/pretimed signal plans
- Actuated (via a ring-barrier graphical user interface)
- User-definable signal control logic through a macro language logic called VAP
- Interfaces to signal controller firmware (virtual controllers) such as Econolite ASC/3 TM SIL or D4
- Interfaces to adaptive algorithms such as Peek's Spot/Utopia, SCATS and SCOOT
- Serial communication to external controllers
- Hardware-in-the-loop connections to VISSIM via NEMA TS2 or TS1 standards, allowing users to connect signal controllers directly to VISSIM

The C-like traffic control macro language, VAP, is supplemented with a flowchart editor for easy data entry, error checking, and debugging. In addition, the ring-barrier controller is used in North America to model actuated signal timings with custom menus to model bus and LRT priority and railroad preemption.

Data provided by the traffic flow model contain detector data which are provided at time steps varying from 0.1 to 1 s. Each detector will provide information of vehicle front end, rear end, vehicle presence, occupancy rate, gap time, vehicle type, and vehicle speed. The controller may utilize only a subset of the detector information. In each controller time step VISSIM contacts all controller units at the end of the current simulation time step. First, the current signalization states and detector data of all signal controllers are passed to the respective tasks implemented as dynamic link library for each controller unit. Second, the tasks are asked to calculate new desired signal states which are subsequently passed back to the traffic simulator. Depending on parameter settings by the controller logic, either these signal states are applied immediately or transition states (e.g., amber) are inserted automatically, as defined in the signal group parameters. In the next simulation time step the vehicles in the traffic simulation will react on the new signalization (Fig. 2.5).

Major signal control manufacturers such as the Netherlands, Haarlem, Peek, Siemens, Swarco, and Thales use this interface to test their controller technology. Besides this universal interface, an additional interface is provided for adaptive controller software. Only one controller task is needed for the adaptive systems SCOOT and SCATS. All detector and signalization data are interfaced with the replication of the central urban traffic control system (UTCS). The UTCS handles the detector and signalization data of all modeled controllers and provides the communication between each individual controller.

The US American market follows the NEMA standard for pretimed, semi-actuated and fully actuated operation. The ring-barrier controller (RBC) interfacing with VISSIM shows fewer limitations than the standard NEMA controller as the

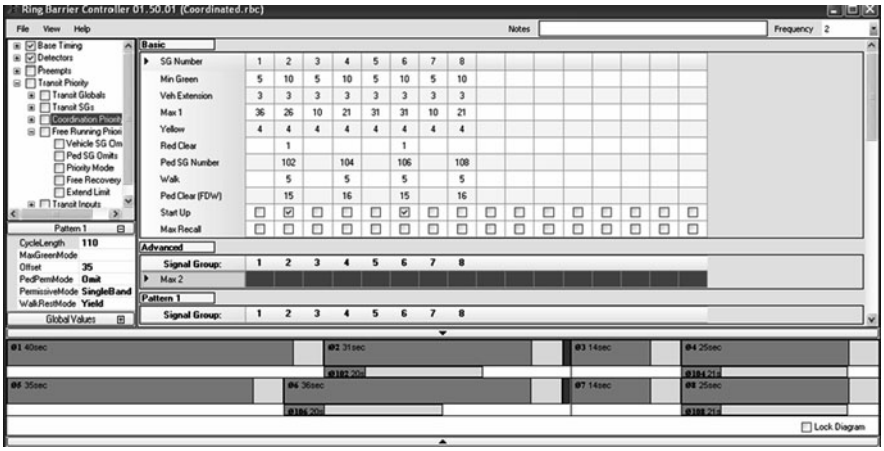


Fig. 2.14 Interface for ring-barrier controllers, a superset of the American NEMA controller

number of signal groups is limited to 16 with 32 vehicle detectors, 16 pedestrian detectors, 2 prioritized preempts, and public transport priority options.

2.5.3 External Driver Model

For users who want to implement their own car-following and lane-changing logic, an interface to external driver models is offered. The external model must be compiled as a dynamic link library. In every time step of the simulation VISSIM will provide the current situation for each vehicle controlled by the external model, i.e., essentially the surrounding vehicles and their attributes. The external model then calculates the acceleration and lane change decision for this vehicle and passes them back to VISSIM. The vehicle is then moved accordingly. The normal VISSIM behavior for the external vehicle is computed as well and provided at the interface in case a user wants to modify only parts of the behavior. A similar interface is available for pedestrian simulation. Network and pedestrian information is provided to the external model and positions and speeds are received from the external model for the next time step.

2.5.4 External Emission Modeling

In order to calculate traffic-related emissions a number of modeling options are available. In German-speaking countries calculations are usually based on the HBEFA (*Handbook Emissions Factors Road Traffic*), which provides emission values for a set of predefined representative driving cycles and vehicle fleets. Also other emission models were developed in the past such as the European COPERT model.

Emission measurements and calculations have been supported by the European project ARTEMIS. The COPERT/CORINAIR methodology allows calculating different pollutants based on measured driving cycles matched with average cycle speeds. Based on intensive measurements on engine test beds, Hausberger (2003) described a detailed instantaneous emission model (PHEM – Passenger car and Heavy duty vehicle Emissions Model). A variety of vehicle and engine types has been embedded by specific engine maps. Driving cycles and engine specifications (Car/lorry, EURO 0 to EURO 4, diesel/gasoline, power, cubic capacity, weight, transmission) are required as input data. Based on a driving cycle, respectively, trajectories with speed and location over time, PHEM calculates engine power required as a result of the driving resistance. Losses in the transmission system and road gradient are being considered. The actual engine speed is simulated by transmission ratios and a gear shift model.

VISSIM provides simulated driving cycles as trajectory data for PHEM, which is calculating the emissions as if the driving cycles are taken from a dynamometer. The toolbox is supplemented by adding various traffic responsive signal control strategies. A study is being conducted comparing the impact of different signal control strategies on traffic-related emissions using the adaptive signal control system MOTION (Fig. 2.15).

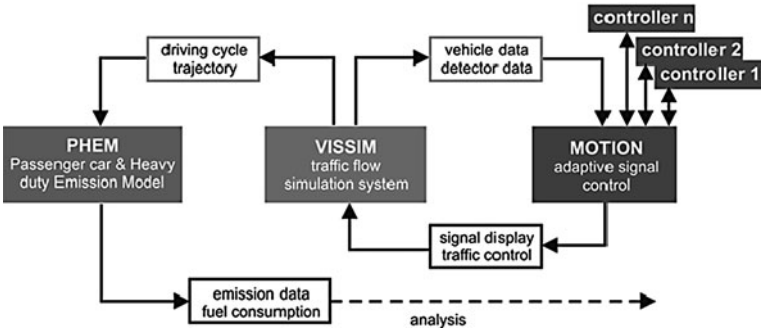


Fig. 2.15 Toolbox for modeling traffic-related emissions in urban environments

References

Benz Th (1985) Mikroskopische Simulation von Energieverbrauch und Abgasemission im Straßenverkehr. Schriftenreihe des Instituts für Verkehrswesen Heft 32, Universität Karlsruhe

Brannolte U (1980) Verkehrsablauf auf Steigungsstrecken von Richtungsfahrbahnen. Forschung Straßenbau und Straßenverkehrstechnik, Heft 318, Bonn

Busch F, Leutzbach W (1983) Spurwechselvorgänge auf dreispurigen BAB-Richtungsfahrbahnen. Auftrag des Bundesministerium für Verkehr, Institut für Verkehrswesen, Universität Karlsruhe

Cascetta E, Nuzzolo A, Russo F, Vitetta A (1996) A modified logit route choice model overcoming path overlapping problems: specification and some calibration results for interurban networks. In: Lesort JB (ed) Transportation and traffic theory. Proceedings from the thirteenth international symposium on transportation and traffic theory, Lyon, France. Pergamon Press, Oxford, pp 697–711

Falkenberg G, Vortisch P (2003) Bemessung von Radverkehrsanlagen unter verkehrstechnischen Gesichtspunkten. Wirtschaftsverlag N.W. Verlag für Neue Wissenschaft, ISBN-13 978-3897019690; Bremerhafen, Germany, 2003

Fellendorf M (1994) VISSIM: Ein Instrument zur Beurteilung verkehrsabhängiger Steuerungen (VISSIM: a tool to analyse vehicle actuated signal control), In: Verkehrsabhängige Steuerungenam Knotenpunkt edited by the German Research Foundation for transport and roads (FGSV), Kassel, p 31–38

Haas M (1985) LAERM Mikroskopisches Model zur Berechnung des Straßenverkehrslärms. Schriftenreihe des Instituts für Verkehrswesen Heft 29, Universität Karlsruhe

Helbing D, Molnár P (1995) Social force model for pedestrian dynamics. Phys Rev E 51(5): 4282–4286

Hausberger S (2003) Simulation of real world vehicle exhaust emissions, VKM-THD Mitteilungen, Heft/vol 82, Graz

Hirschmann K, Fellendorf M (2009) Emission minimizing traffic control – simulation and measurements. Proceedings of mobil.TUM 2009, International scientific conference on mobility and transport, TU München

Hossain MJ (2004) Calibration of the microscopic traffic flow simulation model VISSIM for urban conditions in Dhaka city. Master thesis, University of Karlsruhe, Germany

Hubschneider H (1983) Mikroskopisches Simulations system für Individualverkehr und Öffentlichen Personennahverkehr. Schriftenreihe des Instituts für Verkehrswesen Heft 26, Universität Karlsruhe

Kovali V, Alexiadis V, Zhang L (2007) Video-based vehicle trajectory data collection. Proceedings of the 86th annual meeting of the TRB, Washington, USA, 2007

Kretz T, Hengst S, Vortisch P (2008) Pedestrian flow at bottlenecks – validation and calibration of Vissim's social force model of pedestrian traffic and its empirical foundations. International symposium of transport simulation 2008 (ISTS08), Gold Coast, Australia, 2008

Matsuhashi N, Hyodo T, Takahashi Y (2005) Image processing analysis on motorcycle oriented mixed traffic flow in Vietnam. Proceedings of Eastern Asia society for transportation studies (EAST), vol 5, Tokyo, pp 929–944

Menneni S, Sun C, Vortisch P (2008) Microsimulation calibration using speed-flow relationships. Transportation research board, vol 2088. Washington pp 1–9

Menneni S, Sun C, Vortisch P (2009) Integrated microscopic and macroscopic calibration for psychophysical car-following models. TRB 88th annual meeting compendium of papers DVD, Washington, 2009

Michaelis RM (1963) Perceptual Factors in Car Following. Proceedings from the Second International Symposium on Transportation and Traffic Theory, OECD, Paris, 1965

Reiter U (1994) Empirical studies as basis for traffic flow models. In: Akcelik R (ed) Proceedings of the Second International Symposium on Highway Capacity, vol. 2. Sydney, pp 493–502, ISBN 0869106406

Sparmann U (1978) Spurwechselvorgänge auf zweispurigen BAB-Richtungsfahrbahnen. Forschung Straßenbau und Straßenverkehrstechnik, Heft 263, Bonn

Wiedemann R (1974) Simulation des Strassenverkehrsflusses. Schriftenreihe des Instituts für Verkehrswesen Heft 8, Universität Karlsruhe

Wiedemann R, Reiter U (1992) Microscopic traffic simulation: the simulation system MISSION, background and actual state. In: CEC project ICARUS (V1052) final report, vol 2, Appendix A. Brussels, CEC

Wiedemann R, Schnittger St (1990) Einfluß von Sicherheitsanforderungen auf die Leistungsfähigkeit im Straßenverkehr (Richtungsfahrbahnen). Forschung Straßenbau und Straßenverkehrstechnik, Heft 586, Bonn

Winzer, Th (1980) Messung von Beschleunigungsverteilungen. Forschung Straßenbau und Straßenverkehrstechnik, Heft 319, Bonn

Chapter 3
Traffic Simulation with AVENUE

Masao Kuwahara, Ryota Horiguchi, and Hisatomo Hanabusa

3.1 Introduction

In Japan, traffic simulation modeling was started in 1970s, reasonably earlier period. Two pioneer simulation models developed in University of Tokyo are the input–output method and the block density method in which a highway section is divided into blocks of several dozens to a few hundred meters and flow propagation between the blocks is basically controlled based on the fundamental diagram of the highway. Since most of the earlier models did not have a function of route choice, they were normally applied to a fairly small area including a couple of intersections. Or, whenever they were used for a network-wide simulation, time-dependent diverging ratios were given in advance. Other simulation models in the earlier period in Japan would be MACSTRAN and MICSTRAN by National Research Institute of Police Science. They were mainly used for evaluation of traffic management schemes including traffic signal control.

In the early 1990s, we developed SOUND (Simulation On Urban road Networks with Dynamic route guidance) which has a route choice function and is mainly for a large-scale application such as one to a whole Tokyo metropolitan region. Because of the large-scale application, the flow model in SOUND is quite simplified, for instance, no lane changing, a simple car-following based on a speed-spacing relationship, etc. Although SOUND is very useful to evaluate a network-wide planning and/or management schemes, it has a limitation to reproduce very detailed traffic condition at a smaller area. Therefore, we separately developed AVENUE (Advanced & Visual Evaluator for road Networks in Urban arEas) so as to evaluate more local traffic management strategies.

M. Kuwahara (✉)
Department of Human-Social Information Sciences, Graduate School of Information Sciences, Tohoku University, Aoba 6-6-06, Aramaki, Aoba-ku, Sendai, 980-8579, Japan
e-mail: kuwahara@plan.civil.tohoku.ac.jp

R. Horiguchi and H. Hanabusa
i-Transport Lab. Co. Ltd., 1-4 Kanda-Jinbo-cho Chiyoda-ku Tokyo, 101-0051, Japan
e-mails: rhoriguchi@i-transportlab.jp; hanabusa@i-transportlab.jp

J. Barceló (ed.), *Fundamentals of Traffic Simulation*,
International Series in Operations Research & Management Science 145,
DOI 10.1007/978-1-4419-6142-6_3,

AVENUE is a hybrid traffic simulation in the sense that the flow model is based on the fluid dynamics but the images of displayed vehicles are discrete for easier understanding. AVENUE is normally applied to a small to middle size network such as one with 100 intersections. The flow model in AVENUE is much more detailed way than that in SOUND, and obviously AVENUE also has a route choice function for several different user categories. Since 1992, it has continuously upgraded so as to include additional functions for some special applications explained in this chapter.

Fortunately, AVENUE has a number of practical applications in our country. For example, it has several experiences on impact evaluations of signal control, environmental impact studies, bus transit operations, on-street parking management, road pricing schemes, probe vehicle analyses, and so on.

3.2 Modeling Principles

3.2.1 Hybrid Approach of the Traffic Flow Modeling

There are dozens of microscopic simulation models which have the base on car-following model to allow the flexible modeling of various traffic phenomena. On the other hand, such car-following approach is often criticized because of the difficulty in the parameter calibration and the complexity in the implementation like 'black box.'

Different from many microscopic models, AVENUE employs the 'Hybrid Block Density Method (HBDM)' which combines the fluid approximation of traffic flow and discrete vehicle images (Horiguchi et al., 1994).

In general, from the sake of strict evaluation of the loss caused by traffic congestion, any traffic simulation models are required to reasonably reproduce bottleneck capacity and the dynamics of traffic flow. Since the fluid dynamics of traffic flow may be characterized by the macroscopic 'flow–density' relationship, the fluid approximation approach may take advantage in the implementation with simple but sufficient parameters, such as 'capacity' and 'critical density.'

However, it is clear that the fluid approximation approach cannot respond to the various needs for traffic simulation. AVENUE's HBDM treats vehicles with discrete images to represent individual attributes, such as vehicle type, origin, and destination. A vehicle flows on the road according to the fluid dynamics but can choose lanes and routes according to the individual preferences.

3.2.2 Dynamic Route Choice Model

It is often required for traffic simulation to consider drivers' route choice behavior. AVENUE implements multiple classes of drivers' route choice model according to DUO (dynamic user optimum) and DSUO (dynamic stochastic user optimum)

principles. Those classes are parameterized with 'generic cost function' which considers route length, travel time, number of turns, toll fee, etc.

AVENUE also implements 'fixed-path' routing for the convenience to deal with some special vehicle classes, such as bus and LRT.

3.2.3 Common Framework of Network Traffic Simulation Models

AVENUE is implemented on the common framework of network traffic simulation model to be shared with SOUND or other models. The framework is implemented in object-oriented programming and provides fundamental classes in terms of vehicle generation, network operation, route guidance, traffic signals, result data accumulation, etc. Users can extend the model capability by adding the sub-classes which implement desired functions.

3.2.4 All-In-One Software Package

This is not a modeling principle but a software design policy. Some of the simulation software packages contain several sub-programs, such as 'network builder,' 'signal editor,' and 'animation viewer.' It sometimes increases bothersome for the users by switching, opening, or closing application windows.

Unlike such software, AVENUE packs all the functions into one software package. Users can build simulation data with GUI and continuously verify the data settings with animation in one application window. Figure 3.1 shows the display

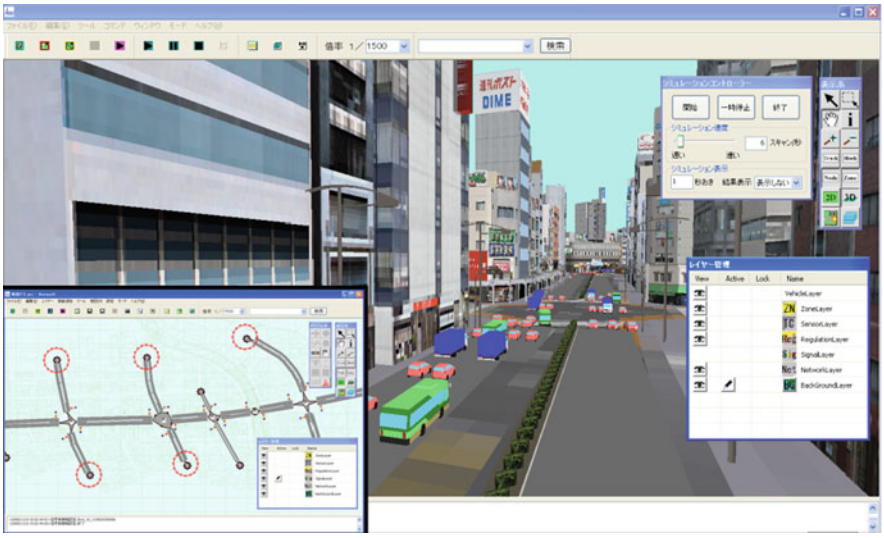


Fig. 3.1 Display of AVENUE Ver. 5

image of AVENUE including the data builder, the data layer manager, and the 3-D animation viewer.

3.3 Traffic Flow Modeling

3.3.1 The Hybrid Block Density Method

AVENUE employs the hybrid representation of traffic flow, which treats traffic as the continuum flow using the block density method (BDM), but at the same time it moves discrete vehicle images for the convenience of the route choice calculation and for handling conflicts of vehicles at intersections and lane change. In this sense, we call the flow model of AVENUE as the Hybrid Block Density Method (HBDM).

In this method, each lane of a link is divided into the series of blocks as shown in Fig. 3.2. The length of each block dL is equal to the distance that a vehicle runs at the free-flow speed V_f of the link during the unit scan interval dt, which is normally set to 1 s, i.e.,

$$\mathrm{d}L = Vf\,\mathrm{d}t \tag{3.1}$$

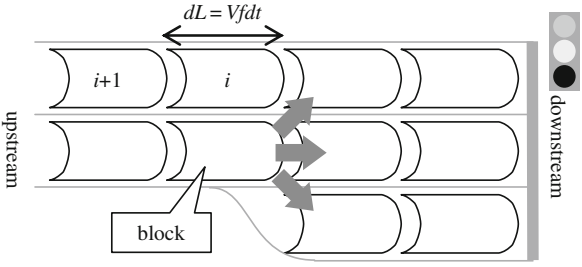


Fig. 3.2 Blocks on each lane of a link

Density of every block is revised based on the flow conservation law and the triangular flow–density ($Q-K$) relationship as shown in Fig. 3.3 so as to be specified by three parameters of jam density K_j, critical density Kc, and maximum flow Qc.

Figure 3.4 illustrates the items used in the flow calculation between the two neighboring block with suffix i and $i+1$ from downstream to upstream. At every scan, the allowable outflow A^{out} and the allowable inflow A^{in} of each block i are calculated from the current density $K(t)$ and the Q–K relationship as follows:

$$A_i^{\mathrm{out}}(t) = \min\left(Kc_i, K_i(t)\right)\frac{\mathrm{d}L}{\mathrm{d}t} \tag{3.2}$$

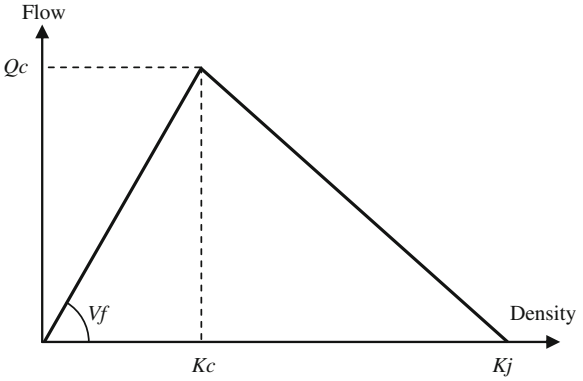


Fig. 3.3 Flow–density relationship assumed in AVENUE

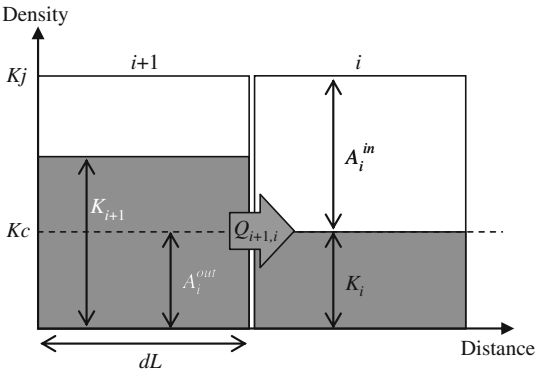


Fig. 3.4 Calculation items in the block density method

$$A_i^{\text{in}}(t) = \begin{cases} (Kj_i - K_i(t)) \dfrac{\mathrm{d}L}{\mathrm{d}t} & (K_i(t) \le Kc_i) \\ \dfrac{Kc_i (Kj_i - K_i(t))}{Kj_i - Kc_i} \dfrac{\mathrm{d}L}{\mathrm{d}t} & (Kc_i < K_i(t) \le Kj_i) \end{cases} \tag{3.3}$$

The flow $Q_{i+1,i}(t)$ between the neighboring blocks from $i+1$ to i is determined by taking the minimum of the allowable outflow of the upstream block and the allowable inflow of the downstream block.

$$Q_{i+1,i}(t) = \min\left(A_{i+1}^{\text{out}}(t), A_i^{\text{in}}(t)\right) \tag{3.4}$$

The densities at the next scan are updated using the inflow and outflow of each block.

$$K_i(t+1)\,\mathrm{d}L = K_i(t)\,\mathrm{d}L + Q_{i+1,i}(t) - Q_{i,i-1}(t) \tag{3.5}$$

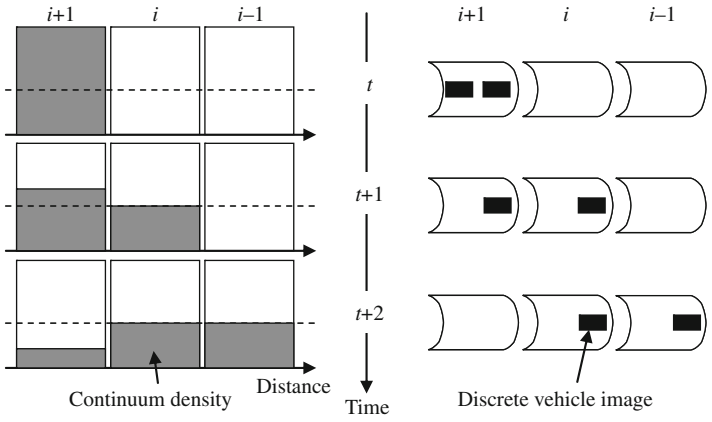


Fig. 3.5 Block traverse of discrete vehicle images in HBDM

As for the extension to HBDM, the blocks have not only the continuum traffic density but also the discrete vehicle images containing the various attributes such as vehicle types, passenger car unit (PCU), destinations, and route choice criteria for the convenience of modeling the vehicle behavior. Figure 3.5 illustrates the movements of discrete vehicles in HBDM. The total PCU of the vehicles moving from block $i+1$ to i is proportional to the flow $Q_{i+1,i}$. The difference between the PCU value and the continuum flow $Q_{i+1,i}$ remains as underflow from $i+1$ to i and will be treated at the next scan interval so that both cumulative values will be equal throughout the simulation period.

3.3.2 Modeling of Lane Choice and Traffic Regulations

As described in the next section, AVENUE incorporates drivers' route choice behaviors to meet the various requirements to traffic simulation. In this modeling, each individual vehicle has its path plan to the destination. This means the vehicle knows its turning movement at every intersection on the path.

Figure 3.6 illustrates the lane choice modeling in HBDM. A link is divided into small blocks of which widths cover one lane. Those blocks are connected in cascade. Therefore a vehicle in the ith block on jth lane, denoted with (i, j), may go forward either of the blocks $(i-1, j+\delta)$, where δ can be either 0 for straight downstream, –1 for left downstream, or 1 for right downstream.

In HBDM, each block has the permissions granting to the acceptable vehicles. In advance to the calculation (h4), a vehicle in the block (i, j) chooses the possible flow to the downstream block $(i-1, j+\delta)$ according to its turning movement, vehicle type, etc. When there are more than two possible flows, the vehicle may choose one of them according to its lane choice preference. Here the preference considers following conditions, as for

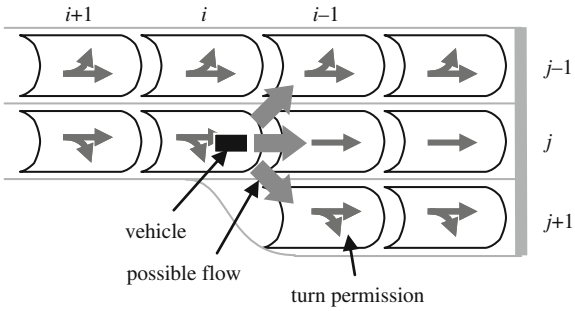


Fig. 3.6 Lane choice modeling in HBDM

- the density of the downstream blocks.
- the queue volume on each lane.
- the distance to the next left/right turn.

In the implementation, the lane choice preference is provided as a fundamental class and the users may extend the preference with their own purpose by defining the sub-class. The preference objects are embedded into each block so that the lane choice behavior of the vehicle may vary according its location (block) and its vehicle type.

3.4 Dynamic Traffic Assignment

3.4.1 Modeling Principal for the Dynamic Route Choice Behavior

AVENUE incorporates the dynamic stochastic user optimal (DSUO) assignment. DSUO is the probabilistic extension of the dynamic user optimal (DUO), which selects the optimum route according to the route cost in the instant it is presented till the user reaches the destination.

The route choice probability in DSUO is given by the following logit choice function.

$$P_j^i(t) = \frac{c_j^i(t)}{\sum_{w \in W_i} c_w^i(t)} \tag{3.6}$$

where

$P_j^i(t)$: the choice probability of the jth route at the current link i on the way to the destination at time t.
W_i: the set of the routes from link i to the destination.
$c_j^i(t)$: the generic cost of the jth route at the current link i at time t.

The generic cost function in AVENUE is described with the following formula, as for

$$c_j^i(t) = \theta \left(a_1 \sum_{l \in L_i} \frac{D_l}{\bar{V}_l(t)} + a_2 \sum_{l \in L_i} T_{l,l+1}(t) + a_3 n_i^{LF} + a_4 n_i^{RT} + \frac{1}{E} \sum_{l \in L_i} F_l \right) \quad (3.7)$$

where

L_i: the set of the links reaching to the destination from the current link i.
D_l: the length of link l (m).
$\bar{V}_l(t)$: the empirical average speed of link l at time t (m/s).
$T_{l,l+1}(t)$: the current travel time from link l to $l+1$ (s).
n_i^{LF}: the number of left turns from link i to the destination.
n_i^{RT}: the number of right turns from link i to the destination.
E: the time equivalent toll fee (yen/s).
F: the toll fee of link l (yen).
θ: the sensitivity parameter of logit choice function.
a_1–a_4: coefficients.

For the users' convenience, AVENUE can treat more than one driver group which have different cost function. It is clear when θ becomes large, the choice behavior closes to DUO.

The empirical average speed of link l is to be outsourced or to be assumed as its free-flow speed. The current travel time from link l to $l+1$ is periodically updated with the average of individual travel time of each vehicle traversing link l to $l+1$ during the time slot from the last update. Every time the current travel time is updated, all the vehicles on the network revise their path plan to the destination.

Since the simulation itself represents the current situation through accumulation of situations at respective time points, the modeling of DSUO seems natural. This can also emulate the advanced travel information provision service in the actual world, such as VICS in Japan.

3.4.2 Dual-Graph Expression for the Route Guidance

In the implementation of DSUO in AVENUE, Dial's assignment algorithm is periodically applied to the 'guidance network,' as shown in Fig. 3.7.

The guidance network is the graph $\boldsymbol{G}^*$ which consists of the set of the origins $\boldsymbol{r}$, the destinations $\boldsymbol{s}$, the guidance vertex $\boldsymbol{v}^*$, and the guidance arc $\boldsymbol{a}^*$. Here, the graph $\boldsymbol{G}^*$ can be deduced as the dual graph of the road network $\boldsymbol{G}(\boldsymbol{r}, \boldsymbol{s}, \boldsymbol{v}, \boldsymbol{a})$, where $\boldsymbol{v}$ is the set of the nodes and $\boldsymbol{a}$ is the set of the links.

For the deduction of $\boldsymbol{G}^*$, two mapping operations are used, those for;

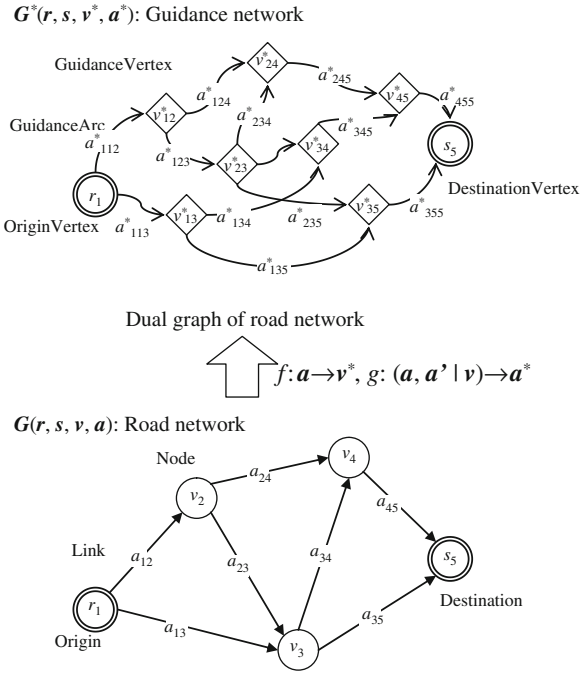


Fig. 3.7 Dual-graph expression of the route guidance in AVENUE

f: the operation to shrink a link a_i into a guidance vertex $v_i{}^*$.
g: the operation to expand a connected link pair a_i and a_j at the node v_k to a guidance arc $a_{ij}{}^*$.

The current travel time in eq. (3.7) is given to each guidance arc $a_{ij}{}^*$. Traffic regulations on the turning movement at an intersection can be considered in the operation g and reflected in the structure of the dual graph $\boldsymbol{G}^*$. Therefore, the Dial's assignment algorithm can consider those regulations in natural way.

3.5 Calibration and Validation

3.5.1 Promotion of Verification and Validation Policy by JSTE

3.5.1.1 Necessity of Standardized Verification and Validation

Any kinds of simulators are requested to reproduce real phenomena as the word of 'simulation' itself means. In traffic simulation, verification and validation of simulation models are therefore mandatory to guarantee that the simulator could be used in practice. Here, 'verification' means the examination if a model follows rules programmed into a computer as the developer intends. For instance, we may examine if

the number of vehicles generated from an origin is exactly the same as the intended value following the probability distribution specified. The shockwave speed produced by a model may be checked if the speed matches with the kinematic theory. On the other hand, 'validation' is the examination if a model reasonably reproduces complex real phenomena. In the real world, we face a lot of traffic phenomena such as merging, diverging, and weaving phenomena, which are quite difficult to describe in a closed form equations. In the validation, we attempt to confirm the reproducibility of real complex traffic phenomena.

The standardization of the verification and validation processes is needed because of the following two reasons:

- a model has to be examined using the common procedure as well as common data. Otherwise, we cannot universally conclude the validity of the model.
- model characteristics can be understood through the results of the verification and validation.

Users often claim that a simulation model contains black boxes they cannot see inside. As the modeling gets complicated so as to reproduce many complex phenomena, the logic inside also gets quite intricacy and becomes a black box. Therefore, one of the ways to disclose the black boxes could be to open the results of the verification and validation. From the results, users could understand properties of models, to what condition the model could be applied, etc.

3.5.1.2 Japanese Verification Manual and Benchmark Data Set for Validation

In Japan, we developed 'Standard Verification Process for Traffic Flow Simulation Model' in 1999 which explains the standard verification process of the simulation models. If you visit the following web site, you could view the proposed process and also take a look at results of verification and validation of several simulation models: <http://www.jste.or.jp/sim/manuals/VfyManE.pdf>.

In our verification process, traffic simulation models are classified into two types: macroscopic and microscopic models. Although the clear definition of these types is not easy, in principle, the macroscopic model means one employing the flow model based on the fundamental diagram, and the microscopic model is one based on the individual car-following behavior. One of the essential differences between the models would be the capacity value of a highway segment. In the macroscopic model, the highway capacity is normally given as the input variable, while, in the microscopic model, the capacity is determined as the result of the simulation based on the individual vehicle behavior.

We proposed the verification process for each of the model types separately on the following properties:

- vehicle generation
- bottleneck capacity

- shockwave propagation
- merging/diverging capacity
- right turn capacity at a signalized intersection due to the opposed traffic
- route choice probability

For vehicle generation, a model is examined if the four points below are properly implemented:

- if the rate of generated vehicles is the same as the input demand rate
- if the probability distribution of the generation pattern is the same as the specified distribution
- if the remaining vehicles that cannot enter the network due to queue backups to the origin are properly accumulated at the origin node

For macroscopic models, since the fundamental diagram is normally given as the input variable, the capacity and the shockwave propagation are examined if they follow the theory. For microscopic models, the fundamental diagram is internally produced. Therefore, they are examined if their values are consistent with the produced fundamental diagram.

For diverging/merging capacity, the capacity value is observed by changing merging/diverging ratio at several different values. The similar examination is requested for the right turn capacity at a signalized intersection by changing the opposed traffic flow rate.

For route choice probability, using a simple network in which the choice probabilities are easily determined based on the theory, we examine if the resulted route choice probability is the same as in the theory.

After the verification explained above, models have to be validated using observed data in the real world. In our country, the benchmark data set has been prepared so that validity of models is quickly examined referring the common observed data. Given the benchmark data set, burdensome field observation is not necessary and also characteristics of each model can be clearly compared with each other. The benchmark data set consists of several different data observed on urban expressways and local streets. For the urban expressways, traffic detector data and the origin–destination volumes between on- and off-ramps are obtained. On the other hand, for local streets, detailed manually observed data including traffic counts at most of the intersections and the origin–destination volumes based on the number plate matching for a couple of study areas are prepared. These data are accessible at <http://www.jste.or.jp/sim/bmdata/index.html>.

3.5.1.3 Verification of AVENUE with Standard Verification Manual

So far, we have verified AVENUE along Standard Verification Process for Traffic Flow Simulation Model (Verification Manual) mentioned above (Horiguchi et al.,

1995). Let us here introduce some of the results to understand the model behavior of AVENUE.

3.5.1.4 Vehicle Generation

Since AVENUE assumes the random arrival pattern at the generation point (centroid), the appearance of the headway of those arriving vehicles theoretically should obey the exponential distribution.

Figure 3.8 shows the headway distribution of the generated vehicles when the generation flow rate was set to 2000 (veh./h). The graph contains the results of five trials with different random number series. In this case, AVENUE always generated 2000 vehicles within an hour. Although the headway distributions were slightly different depending on the random series, they fairly followed the theoretical exponential distribution curve. Only the frequency of the short headway range as 0–1 s was lower than the theoretical value, but this could be explained that the minimum headway would be regulated by the capacity of the link discharging from the generation point.

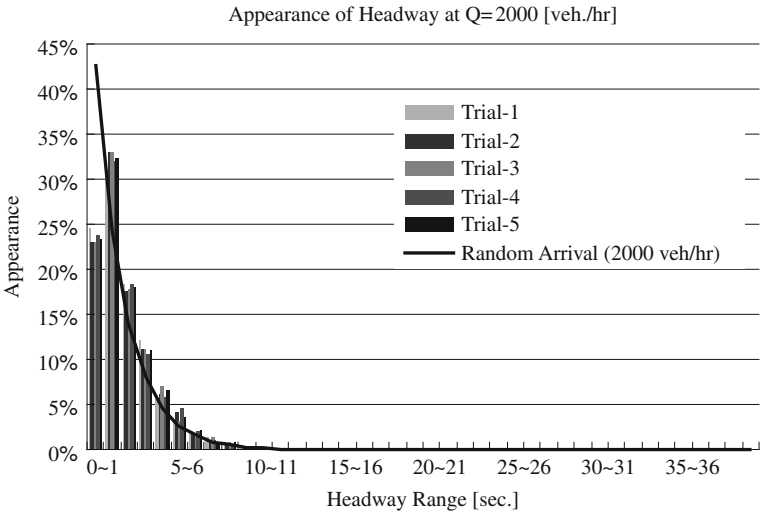


Fig. 3.8 Headway appearance of the generated vehicles in AVENUE

3.5.1.5 Shockwave Propagation

When a queue beginning at a bottleneck grows up to the upstream link, traffic that need not pass through the bottleneck is also affected. Consideration of this phenomenon requires handling of the physical queue to maintain the density of congested flow according to the appropriate flow characteristic during simulation.

In the case of physical queue, the speed of shockwave propagation can be expressed by the relationship between the demand arriving from the upstream side and the bottleneck capacity while using the shockwave theory. The Verification Manual requires to compare the theoretical flow state transition and the simulation result with the input conditions as illustrated in Fig. 3.9.

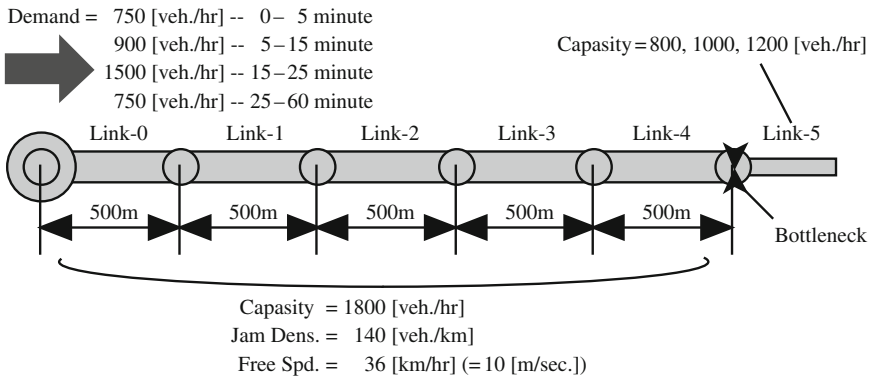


Fig. 3.9 Input settings in the verification procedure for shockwave propagation

In Figure 3.10, the graph above shows the cumulative traffic counts at the upstream end of each link from the simulation result, while the diagram below illustrates the theoretical state transition in time space. There are five forward waves (FW1–5) and one backward wave (BW1) in the diagram. When BW1 crosses the end of each link, we can see the tangent of the cumulative curve of the link declines from the arriving demand rate to the bottleneck capacity. Similarly, when FW5 crosses the link end, the flow rate recovers to the arriving demand rate.

3.5.1.6 DSUO in the Route Choice Model

The simulation model based on DSUO adopts a framework in which users select routes on the basis of presented route cost. This type of model is often used to evaluate the dispersing of traffic spatially by means of information services and road construction.

The verification of DSUO is made using a simplified network with two routes, as shown in Fig. 3.11, because of the increased difficulty of determining flow patterns to achieve DSUO as the network becomes complicated.

Figure 3.12 shows the cumulative flows of links 1 and 3, which are on the alternative routes. In this case, the logit sensitivity parameter θ was set to 0.01 and the generic cost function only considered the current travel time, where (a1, a2, a3, a4) = (0, 1, 0, 0) in eq. (3.7). Namely, the choice probabilities of those alternative routes can be calculated only from the time difference between two routes, as shown in Fig. 3.13.

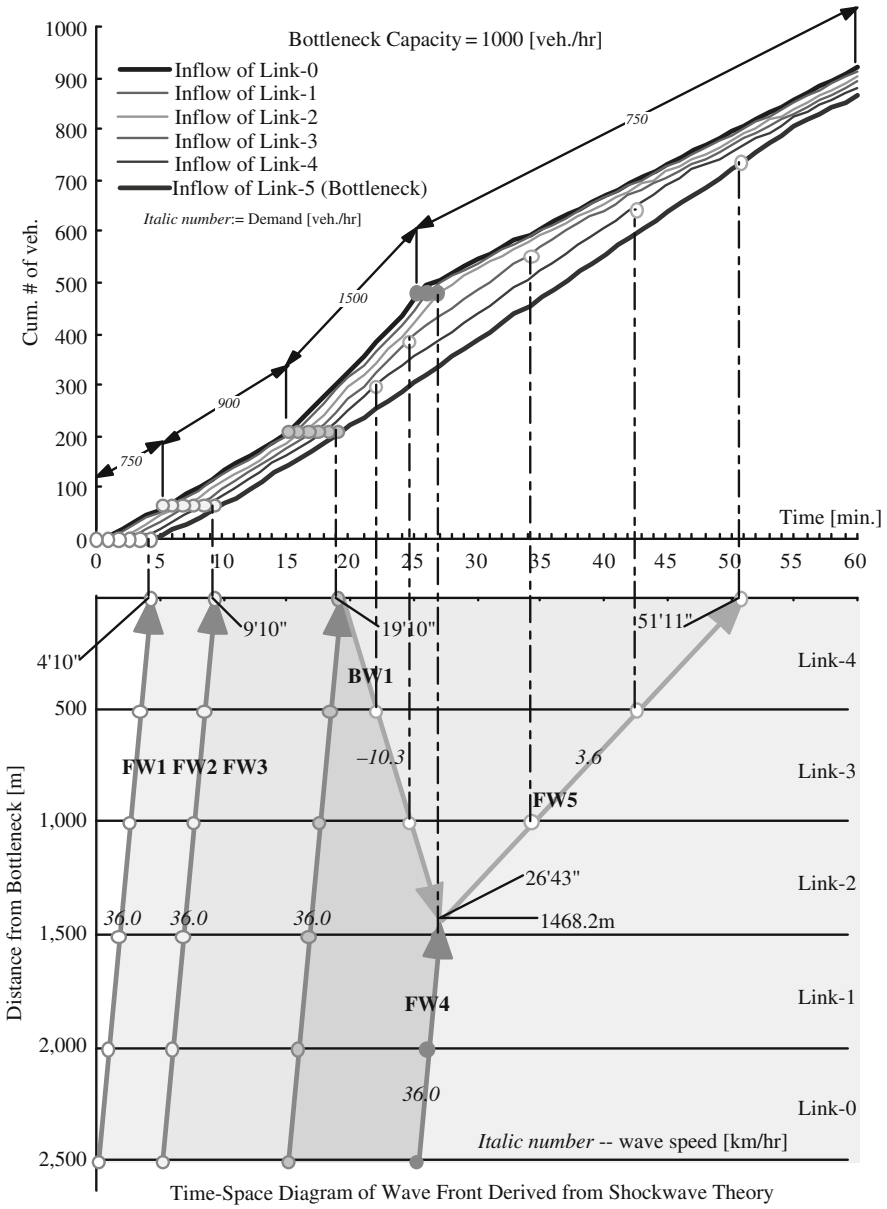


Fig. 3.10 The result of the verification for the shockwave propagation in AVENUE

From Fig. 3.11, about two-third vehicles selected link 1 at the beginning when the cost difference was equal to 100 s in free-flow condition. Then, as the queue on link 1 grew heading to the merging point of link 1 and 3, the cost difference became small, and the traffic flows to both routes became almost equivalent at around 0:15 to 0:20.

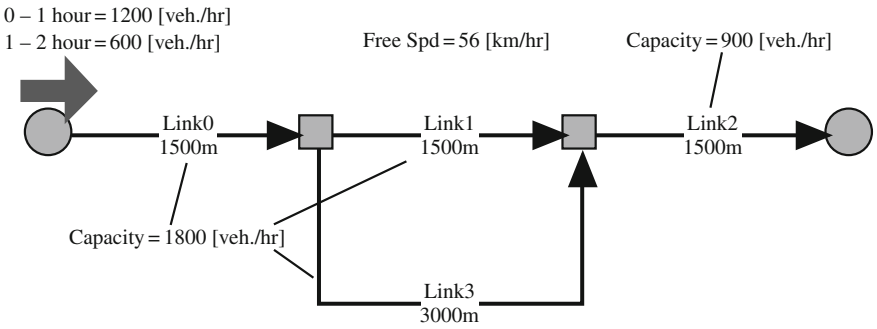


Fig. 3.11 Input settings in the verification procedure for DSUO

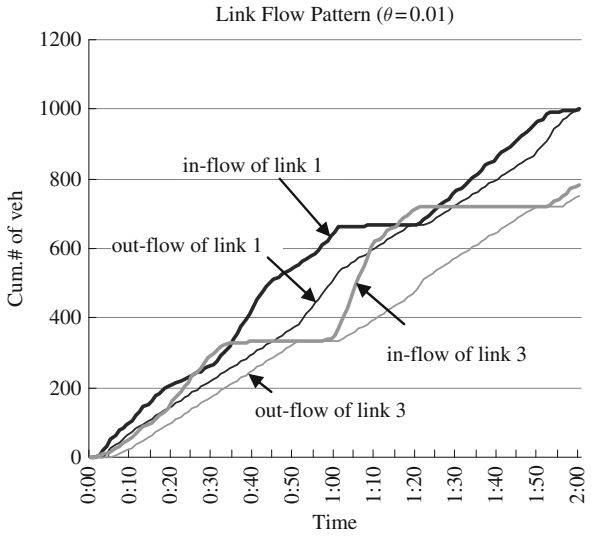


Fig. 3.12 The result of the verification for DSUO

It is theoretically said that DUO/DSUO cannot achieve DUE (dynamic user equilibrium), since the present route cost may not represent the actual cost that the drivers will experience from now. Therefore, the simulation result was not stable after 0:20 when the costs of two routes were equivalent and revealed its behavior as so-called hunting phenomenon, which was intrinsic in DSUO.

3.5.2 Validation of AVENUE with Standard Benchmark Data Set

As described in Section 3.5.1, not only the verification of simulation models but also the validation can give us useful information concerned with the models' nature.

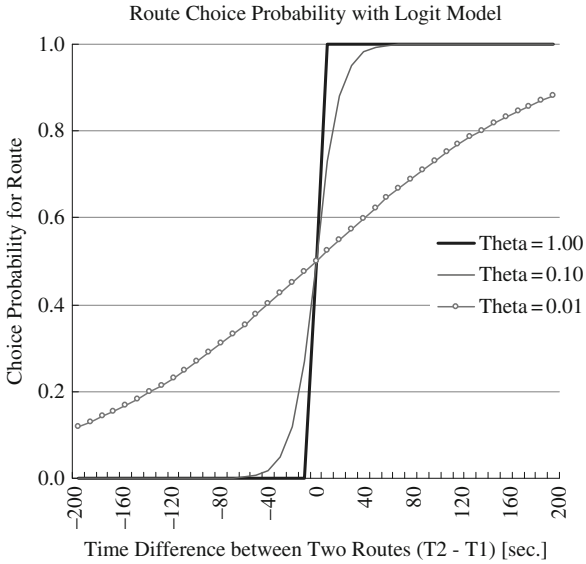


Fig. 3.13 Route choice probability with Logit model

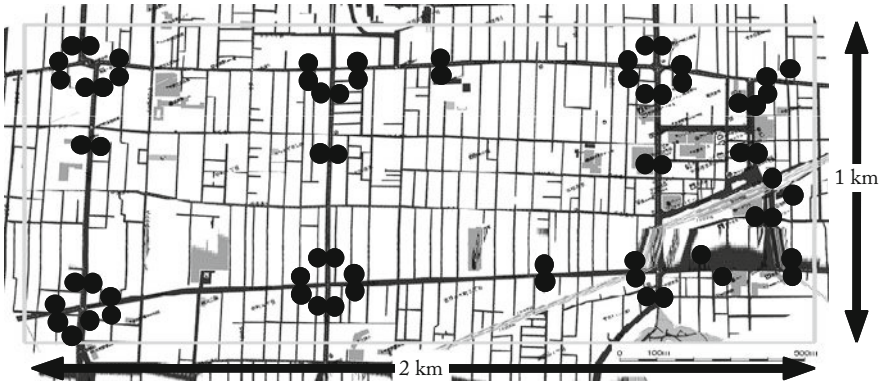


Fig. 3.14 Subject area and the surface street network in Kichijoji–Mitaka BM

We have so far validated AVENUE through the application with 'Kichijoji–Mitaka Benchmark Dataset (BM)' (Hanabusa et al., 2001) for the network of surface streets.

Figure 3.14 illustrates the surface street network in Kichijoji–Mitaka area in Tokyo, on which precise OD trips were collected by recording the plate numbers of every vehicles at the marked points in the figure, as well as travel times and signal timings. The route choice behavior in this area was also identified with disaggregated logit choice analysis.

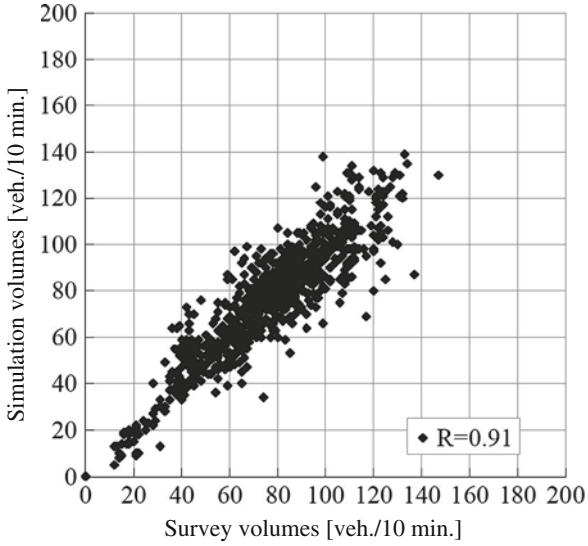


Fig. 3.15 Volume comparison in the validation with BM

Figure 3.15 compares the link throughput volumes in every 10 min from the simulation result with survey data. In the validation, AVENUE was applied to the whole network, then it gave quite satisfactory result as $R^2 = 0.91$.

Figure 3.16 shows the comparisons in average travel speed along some major streets across the area. Although the simulation results did not perfectly follow the

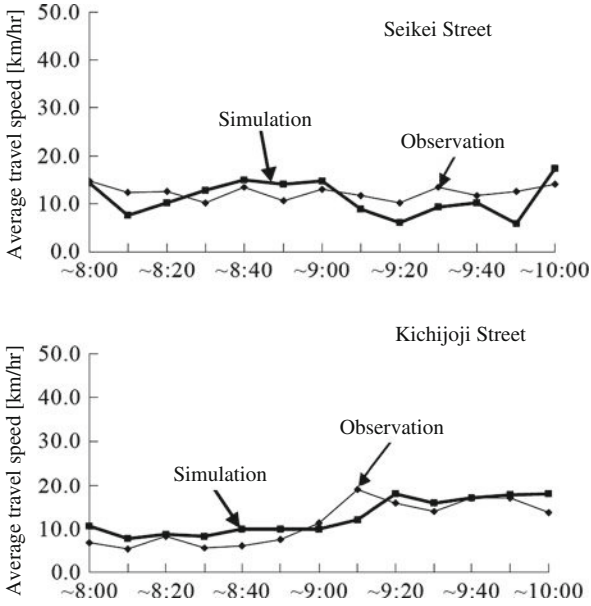


Fig. 3.16 Travel speed comparison in the validation with BM

observation, it seemed to fairly reproduce the changes in travel speed in terms of the speed range and the tendency of the changes.

3.6 Extended Modeling Capabilities: Working with External Applications

3.6.1 Time-Dependent OD Estimation

3.6.1.1 Outline of the Method

Time-dependent OD (origin–destination) volumes are always burdensome to be acquired for the input data of simulation models. Conventionally, the OD volumes have been observed based on questionnaire surveys once every 5 years or so. However, for the input OD of simulation models, we need more time-dependent for smaller zones.

This study develops the model to estimate time-dependent OD volumes in a general network, which consists of links and nodes (Oneyama and Kuwahara, 1996). In this model, each link has time-dependent link travel time which is however flow independent and must be input in advance based upon the field observation. The model consists of two parts: (1) construction of the relationship between the time-dependent OD volumes and traffic counts at links and (2) estimation of a unique time-dependent OD matrix.

3.6.1.2 Relationship Between OD Flow and Link Flow

In the first part of the model, the relationship between OD flow and link flow is established by introducing a route choice probability determined from the given time-dependent link travel times. Time axis is divided into discrete time intervals of equal length Δt, and time interval h is defined as time interval $[h\Delta t, (h+1)\Delta t]$. $Ta(h)$, travel time at link a at time interval h, is assumed to be a multiple of an integer Δt. Hence, Δt must be sufficiently small so that change of link travel time over time can be well described. It is also assumed that link travel time, $Ta(h)$, link flow at link a at time interval h, $v_a(h)$, and OD flow departing from origin node of OD pair w at time interval h_r, $q_w(h_r)$, do not vary during each time interval, which means that they stay constant values at the start of a time interval. Then, the relationship between OD flow and link flow is written as

$$v_a(h) \cdot \Delta t = \sum_w \sum_{h_r} p_{aw}(h_r, h) \cdot q_w(h_r) \cdot \Delta t \tag{3.8}$$

where $p_{aw}(hr, h)$ = probability that a vehicle departing from origin node r of OD pair w during time interval hr enters link a during time interval h. Since time-dependent link travel times are given and users are assumed to choose their routes

only based on route travel times, probability $p_{aw}(hr, h)$ can be estimated from a route choice model you would like to employ.

3.6.1.3 Estimation of Unique OD Matrix

In the second part, a time-dependent OD matrix is uniquely estimated by applying the entropy maximizing method under constraints of the relationship between OD and the observed link flows obtained in the first part. Unknown $qw(hr)$ must be estimated so that eq. (3.8) is satisfied. The number of conditions eq. (3.8) is the number of observed link (a) times the number of observed time intervals (h), which is normally less than the number of unknowns, the number of OD pairs (w) times the number of time intervals (hr). Hence, we can find many sets of OD matrices which satisfy eq. (3.8) and the problem is how we should choose a unique matrix among the candidate matrices satisfying eq. (3.8). In this model, we apply the entropy maximization method to this time-dependent model in order to choose a unique matrix. Here, a prior OD flow departing from the origin r of OD pair w at time interval hr is denoted as $\hat{q}_w(h_r)$. Then, OD flow $q_w(h_r)$ is estimated as

$$q_w(h_r) = \hat{q}_w(h_r) \prod_{a,h} X_a(h)^{p_{aw}(h_r,h)} \tag{3.9}$$

where $X_a(h)$ is a parameter obtained by solving the following equation:

$$v_a(h) X_a(h)^{-1/\gamma} = \sum_{w,h_r} \left\{ p_{aw}(h_r, h) \hat{q}_w(h_r) \prod_{a,h} X_a(h)^{p_{aw}(h_r,h)} \right\}, \quad \forall a, h \tag{3.10}$$

Another parameter γ describes the degree of stochastic link flow variation due to the measurement error. As γ gets larger, the stochastic variation gets smaller, and for $\gamma =$ infinity, the variation becomes zero with no measurement errors.

In this way, a unique OD matrix is estimated from traffic counts and link travel time. As seen, this method can incorporate some a priori knowledge on OD volumes. For instance, we may have several in advance qualitative knowledge such that some OD volumes are definitely larger than others, OD volume A is smaller than OD volume B, etc. This a priori knowledge could be flexibly incorporated in the estimates even if the knowledge is not quantitative. Also, we may supply such a priori OD knowledge in the aggregated zone level, which could be known from some existing OD survey with larger zone sizes. Although the method requires observed time-dependent link travel times, they could be often measurable from traffic detectors and/or probe vehicle information, etc., in these days.

3.6.1.4 Application to Tokyo Metropolitan Expressway

The proposed method was applied to the OD estimation of the Tokyo Metropolitan Expressway, since the true OD volumes were measured by the sampling survey with

about 8% sampling rate. The hourly OD volumes were estimated given the daily OD volumes and the hourly on-ramp volumes as a priori information. Figure 3.17 shows the relationship between the hourly estimated and observed on-ramp to off-ramp OD volumes. The correlation coefficient of 0.87 looks reasonable, but the RMS error of 53 (veh./h) was still considerably large. The main source of the error would be the route choice probability. In this study, we simply assumed that everyone chooses his/her route based on route travel times. This assumption however may not sufficiently describe user route choice behavior.

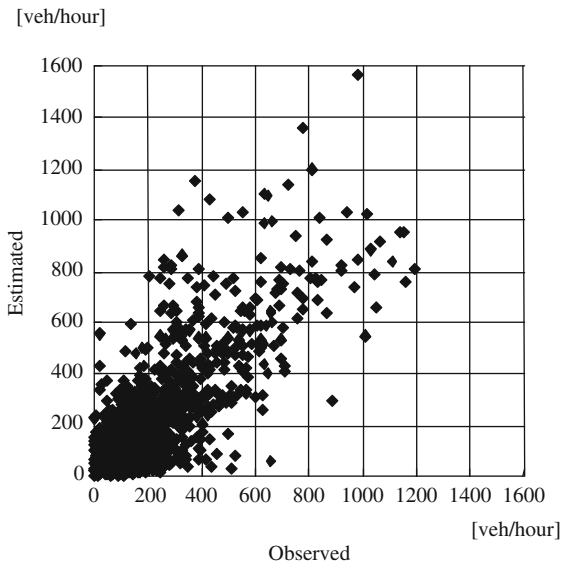


Fig. 3.17 Estimated and observed hourly OD volumes for the Tokyo Metropolitan Expressway

3.6.2 Automatic Parameter Tuning

3.6.2.1 Introduction

Every traffic simulator has some parameters inside the model and they must be calibrated so as to reproduce the real phenomena you study. Normally however, the parameter calibration has been done manually by running the simulation many times. An expert of the simulator may spend substantial time and effort to tune the parameters until the output becomes satisfactory level. Ways of parameter tuning totally depends on the expert and hence different parameter values could be resulted even with the same model and the same data. Therefore, we need the more systematic methodology to calibrate parameters.

We proposed an algorithm which automatically calibrates parameters of macroscopic simulation models so as to reproduce both link travel time and link volume that reasonably agree with the real observed data. Since link capacities are the most

influential parameters in the macroscopic model, a method that automatically tunes the capacity values is briefly introduced here.

The basic issue in the parameter calibration is how to revise the parameter values based on the present output of the simulator so as to improve the agreement with the observed traffic condition. We need to have a method that finds the descendent direction of the parameters to minimize the objective function representing the discrepancy between the simulated and observed traffic conditions. Furukawa and Kuwahara (2000) proposed a systematic tuning method in which the objective function was defined as the difference between observed link travel time and simulated one and the point queue model was employed to find out the descendent direction of parameters. Because of the point queue model, the computation time was greatly saved so that the algorithm could be applied to a large-scale network such as the metropolitan expressway network. However, the calibrated result was not always satisfactory. Therefore, instead of using the point queue model, we proposed another method to find the descendent direction based on traffic engineering expert knowledge (Kuwahara and Oneyama, 2003). The method has been tested in both hypothetical and real networks.

3.6.2.2 Tuning Parameters to Reproduce Link Travel Time

The link capacity is defined by turning directions as follows:

μ_i^k = capacity of link i for turning direction k,
$k = 1$ (through), 2 (left turn), 3 (right turn).

Link capacity values are calibrated so as to reproduce not only link travel time but also link traffic volume.

Even if travel time on link i is not the same as the observed travel time, we cannot immediately judge whether the link capacity should be modified because the link travel time may not depend only on capacity of link i. When simulated link travel time is longer than the observed one, one of the followings could be the reason:

(a) the capacity value of link i is too small,
(b) a queue backs up from downstream due to a too small link capacity value of a downstream link, or
(c) higher flow rate arrives at link i due to a too large link capacity of an upstream link.

Among these three cases, modification of link i capacity is only appropriate for case (a), but downstream and upstream link capacities should be modified for cases (b) and (c), respectively. Only in case (a), link i is a bottleneck link at the head of the queue. As seen here, we should modify capacity values of only bottleneck links, since other link capacities do not directly influence on traffic condition. Focusing on bottleneck links, the updating strategy is proposed for the following four combinations.

For [1] in Table 3.1, link i is a bottleneck link both in observation and in simulation. The capacity value is therefore modified so that the degree of congestion upstream agrees with the observation. For [2], since bottleneck link i in observation is not a bottleneck in the simulation, the capacity value should be reduced so as to be the bottleneck. For [3], link i is a bottleneck link only in the simulation but not in the observation. Two alternative ways are possible for the modification: (A) increase link i capacity and (B) reduce capacity of upstream link so that lesser flow rate arrives at link i. In this research, we employ '(A) increase link i capacity' for the following reason. If link i capacity is increased, the degree of congestion heading from link i would gets smaller and case [3] would be expected to shift to case [2]. Thus, even if the right modification could have been (B) instead of (A), we could implement (B) after traffic condition shifts from [3] to [2] by the (A) operation.

Table 3.1 Observed vs. simulated traffic conditions

	Bottleneck link in simulation	non-bottleneck link in simulation
Bottleneck link in observation	[1] Link capacity should be modified so that the upstream congested condition is agreed with the observed condition	[2] Decrease the link capacity so that the link becomes a bottleneck link
Non-bottleneck link in observation	[3] Increase the link capacity so that the link does not become a bottleneck link	[4] No modification is required

3.6.2.3 Tuning Parameters to Reproduce Link Traffic Volume

Observed and simulated traffic volumes are denoted as

f_{ik}^{sim} = simulated traffic volume on link i for turning direction k.
f_{ik}^{obs} = observed traffic volume on link i for turning direction k.

The link capacity value is modified as shown in the Table 3.2:

Since simulated flow f_{ik}^{sim} cannot exceed capacity μ_i^k, always $\mu_i^k \geq f_{ik}^{\text{sim}}$, only three combinations above are possible. For (III), since capacity μ_i^k is smaller than observed volume f_{ik}^{obs}, capacity μ_i^k should be obviously increased to let simulated volume be closer to the observed volume. On the other hand, for (I) and (II), both observed and simulated volumes are smaller than capacity μ_i^k. For (II), if capacity

Table 3.2 Modification of link capacity

(I) $\mu_i^k \geq f_{ik}^{sim} > f_{ik}^{obs}$	Decrease link capacity
(II) $\mu_i^k > f_{ik}^{obs} > f_{ik}^{sim}$	Increase link capacity
(III) $f_{ik}^{obs} > \mu_i^k > f_{ik}^{sim}$	Increase link capacity

μ_i^k is increased, simulated volume may get larger and consequently condition (II) would shift to (I). While for (I), reduction of capacity μ_i^k would shift condition to (II). As a whole, the above modification strategy shifts condition from (I) to (II) or from (II) to (I). By this modification, f_{ik}^{sim} is expected to eventually converge to f_{ik}^{obs}.

3.6.2.4 Range of Capacity Value and Step Size of Updating

From the highway geometry, we approximately decide the range of each capacity value. The automatic calibration therefore should be made within the range based on the geometry. The step size of updating capacity value at the nth iteration $\Delta\mu_i^k$ is designed as follows so that the step size gradually gets smaller.

$$\Delta\mu_i^k = \Delta\mu^*/n, \tag{3.11}$$

$\Delta\mu_i^k$ = step size of updating capacity value at the nth iteration.
$\Delta\mu^*$ = initial step size.

As explained above, we employ an empirical tuning methodology based upon traffic engineering knowledge to iteratively update the parameter values so that both observed link flows and link travel times are well reproduced in the simulation model. From an application to a hypothetical network composed of 104 links, the proposed method seems quite permissible to adjust the parameters.

3.6.3 Valuation Platform of Vehicle Probe Information System

AVENUE was used in 'HAKONIWA' (Hino et al., 2004), the virtual experiment tool for the Internet CAR Project (Uehara et al., 1998), which have been operated from 2000 (http://www.internetits.org/). The Internet CAR Project aims to create ITS platform which all vehicles are connected through the Internet. By constructing road–vehicle–human network, new application and services can be developed in open platform.

HAKONIWA, meaning small boxed garden in Japanese, consists of three simulation modules, the traffic simulator (AVENUE), the weather condition emulator, and the wireless communication simulator, as shown in Fig. 3.18. Those simulators provide the internet communication interface with IPv6, and every sensors on each vehicle are given unique IP addresses. Therefore, any on-board application can retrieve necessary sensor data through the internet in HAKONIWA. For instance, AVENUE emulates GPS and speed sensors of individual vehicle.

Since the server-side software for probe information system can collect necessary probe data through the internet as well, vehicle probe applications can be developed in the close-to-real situations using HAKONIWA.

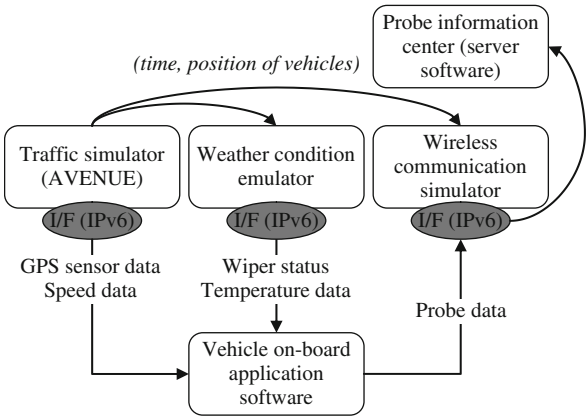


Fig. 3.18 Framework of HAKONIWA

For a demonstration purpose, a sample on-board application which simply communicates with the center server on the Internet has been developed on HAKONIWA. In the validation, 15 Wi-Fi access points are configured within the subject area of which size is approximately 10×20 km.

Figure 3.19 figures the trajectories of 100 probe vehicles running in the area. The round-trip time (RTT) of the communication packets between those vehicles and the center server have been evaluated. Figure 3.20 shows the RTT of a vehicle which varies in time. This means the communication performance may be reproduced by HAKONIWA depending on the position of each vehicle.

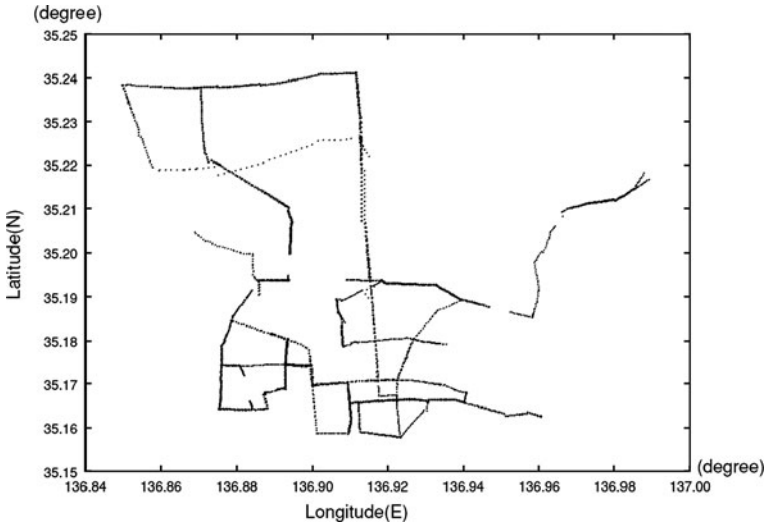


Fig. 3.19 Probe vehicle trajectories in HAKONIWA

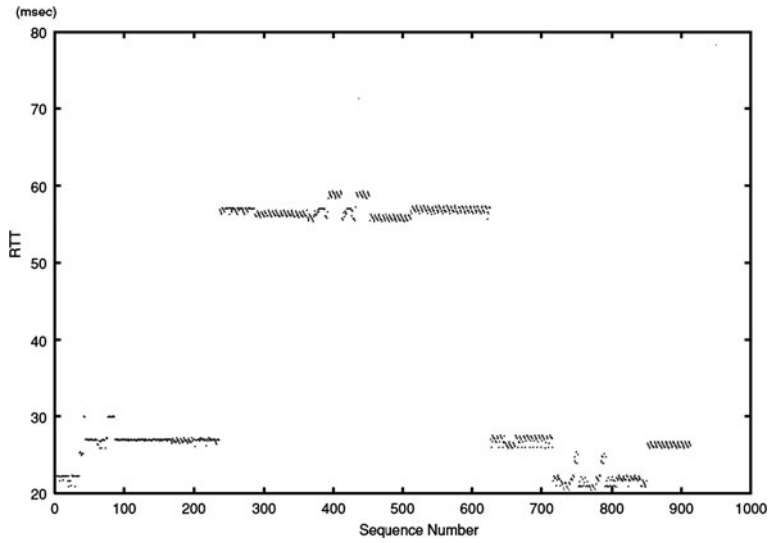


Fig. 3.20 Changes in the round-trip time spent for the vehicle-to-center communication

3.6.4 Valuation Platform for Adaptive Signal Control System

AVENUE provides the emulation function of various traffic sensors, such as ultra-sonic detectors, image sensor detectors, AVIs (automatic vehicle identifier), and infrared beacons used in VICS in Japan. The emulation function will provide the sensor data, as listed below, to the system outside of AVENUE with inter-process communication at every second.

- Ultra-sonic detector
  - cumulative counts of passing vehicles.
  - occupancy.
- Image sensor detector
  - cumulative counts of passing vehicles.
  - section speed/queue length.
- AVI
  - plate number of each passing vehicle.
- Infrared beacon
  - ID of VICS on-board unit.
  - ID of the previous beacon.
  - Time stamp at the position of previous beacon.

Adding to the sensor emulation, AVENUE accepts the update command for the duration of each signal phase from the outside system. With those functions, AVENUE can be utilized as the virtual experiment field of advanced signal control system.

Asano et al. developed the adaptive signal control algorithm 'CARREN' with direct measurement of total delay using AVIs installed besides the intersections (Asano et al., 2003). CARREN dynamically updates signal parameters cycle-by-cycle to minimize the total delay caused by the signal control, so that it is expected to reduce the loss when the traffic situation will drastically change including incident scene. As CARREN had been planned to implement as a pilot system in the real world, it was necessary to check its behavior and its performance in advance through the virtual experiment on AVENUE.

The study network was selected along the Nagakute line in the Nagoya central district. The network has a tree shape and consists of 10 links and 11 intersections. The places of the sensors are shown in Fig. 3.21, according to the actual place of the sensors.

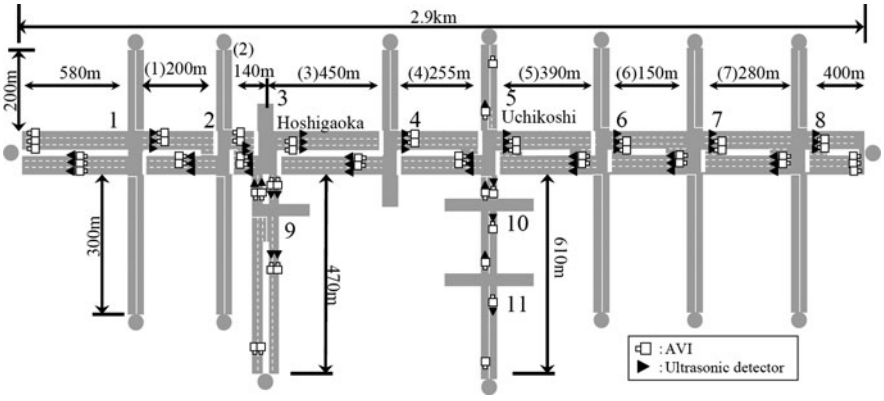


Fig. 3.21 Study area for the virtual experiment of CARREN

In the virtual experiment, three cases during 2 h in the peak period were examined whether parameters can follow the change of traffic demand.

The saturation degrees of the intersections stayed between 0.5–0.7 for case 1 and 0.5–0.95 for case 2. In case 3, the degrees of saturation were the same as case 1, but the saturation flow rate decreased 20% in the latter half of the simulation time, by assuming an incident.

The simulation results were evaluated in terms of total delay of each intersection. Figures 3.22, 3.23, and 3.24 show the changes in the total delay of the major roads and of the minor roads at Uchikoshi and Hoshigaoka intersection in each case. In those figures, 'without algorithm' means the case that the signal parameters would not change from the initial settings during the whole simulation period.

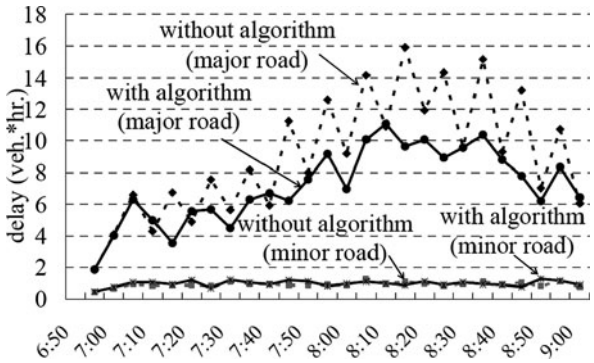


Fig. 3.22 Changes in the total delay in case 1

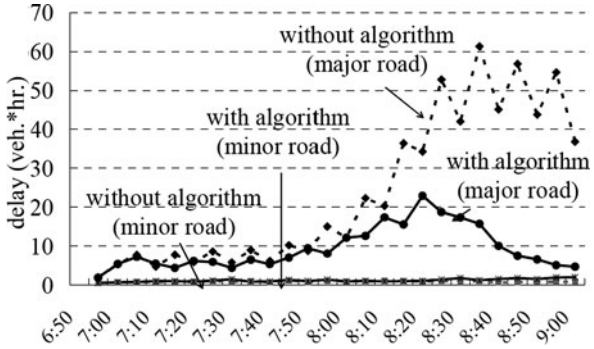


Fig. 3.23 Changes in the total delay in case 2

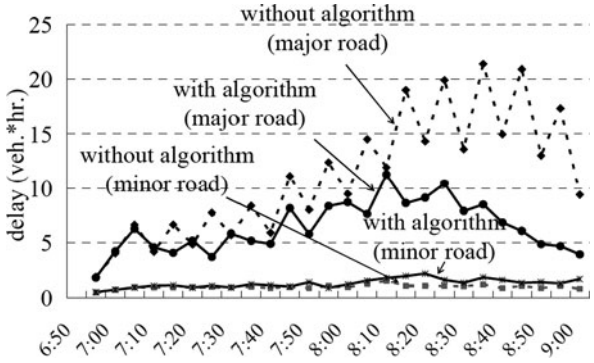


Fig. 3.24 Changes in the total delay in case 3

Here, the delays decreased in every case. Especially in case 3, although the delay without this algorithm increases compared to case 1, the delay is almost the same as case 1 with the system. It follows from this that the algorithm can control considering the decrease of the saturation flow rate by measuring not only the arrival flow but also the departure flow.

3.7 Selected Overview of Advanced Case Studies and Applications – Estimation of City-Scale Noise Level Distribution from Road Traffic

Environmental protection especially global warming is now the hottest international issue, and reduction of CO_2 emission from transport sector is the assignment seriously requested to each of motorized countries. On the other hand, noise abatement is also not negligible issue, since our human beings directly sensate noise and noise exposure could cause not only physical but also emotional health problems such as annoyance, sleep disturbance, hypertension, and so on.

In this section, an approach of estimation of road traffic noise and CO_2 emission in citywide area linked with traffic simulation is introduced (Tsukui et al., 2009). In general, there are many combinations of traffic simulation and environmental models to evaluate sound power levels and CO_2 emission. But we have to select a combination carefully which can suitably evaluate the output. Considering these model characteristics, traffic simulator AVENUE, ASJ Model 2003 (Kono et al., 2004), and the average travel speed-based CO_2 emission model, proposed by National Institute for Land, Infrastructure and Transport, Ministry of Land, Infrastructure, and Transport, NILIM Model (SAE 1982), are chosen.

ASJ Model and NILIM Model can reasonably respond to dynamic traffic conditions and request only inputs obtained from traffic simulator AVENUE. In ASJ Model, sound power levels (L_{WA}) both for steady running condition and transient running condition can be calculated as shown in Fig. 3.25. Note that both equations are the function of only vehicle speed V.

On the other hand, NILIM Model evaluates CO_2 emission using average travel speed of a vehicle within one trip. The equation of the model is

$$\mathrm{EF}_{\mathrm{CO}_2}(v) = a_1 / v + a_2 v + a_3 v^2 + a_4$$

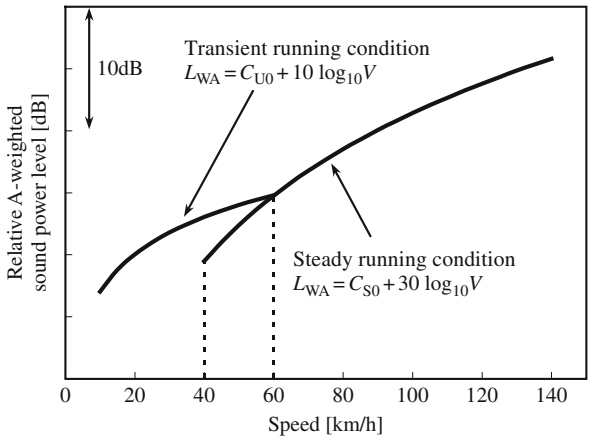


Fig. 3.25 Sound power calculation by ASJ Model 2003

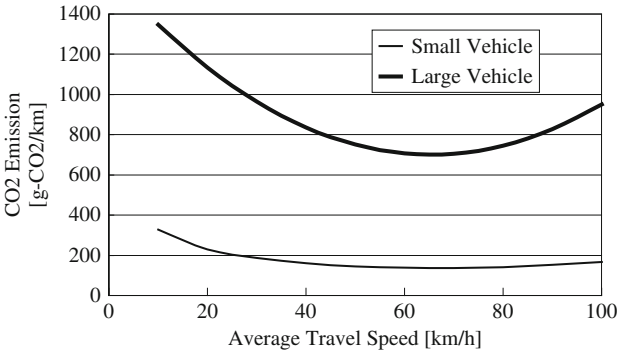


Fig. 3.26 CO_2 emission model used in the study

where EF_{CO_2} is emission factor of CO_2 per unit distance (g-CO_2/km), v is average travel speed of a vehicle within one trip (km/h), and a_1, a_2, a_3, and a_4 are parameters. Figure 3.26 shows the relationship between CO_2 emission and average travel speed for small and large vehicles, respectively.

As a case study, a suburban city in Japan 'Tsukuba city' was selected. Tsukuba city is located about 60 km away from Tokyo in the direction of northeast. There are two major roads, the national highway and the Joban expressway. The target size for simulation in Tsukuba city is 7 km radius. Figure 3.27 shows one frame of the simulated traffic flow animation and estimated road traffic noise in Tsukuba city. According to the observed traffic flow of 12 intersections and travel time in main road sections, we confirmed the estimated traffic situation seems to reasonably reproduce the citywide traffic condition in Tsukuba city. From Fig. 3.27, it is found that the noise levels along the main urban roads and the expressway are high. And the CO_2 emission from whole road network was also calculated by introducing the

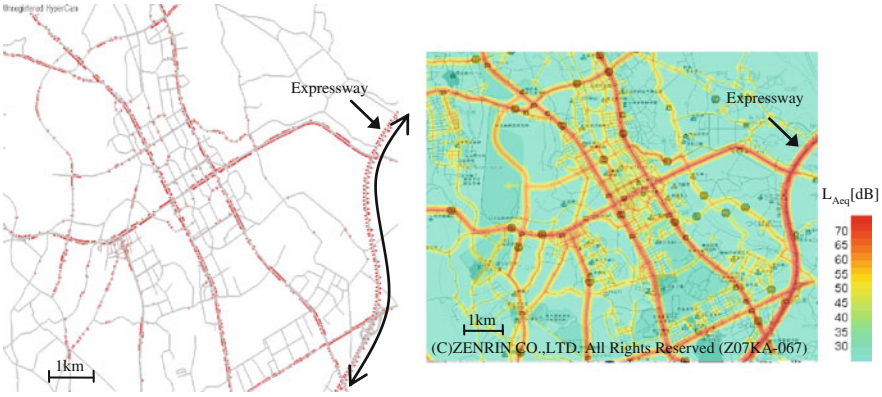


Fig. 3.27 Simulated traffic flow (*left*) and estimated road traffic noise (*right*)

simulated traffic flow into our CO_2 evaluation method. As a result, the CO_2 emission in the study area of Tsukuba city was estimated to be 43.1 ton CO_2/h.

As an example of the traffic flow managements, we examine the changes of the traffic noise and CO_2 emission caused by the new road infrastructure, which is shown in Fig. 3.28 in bold lines. Figure 3.29 shows the noise maps estimated

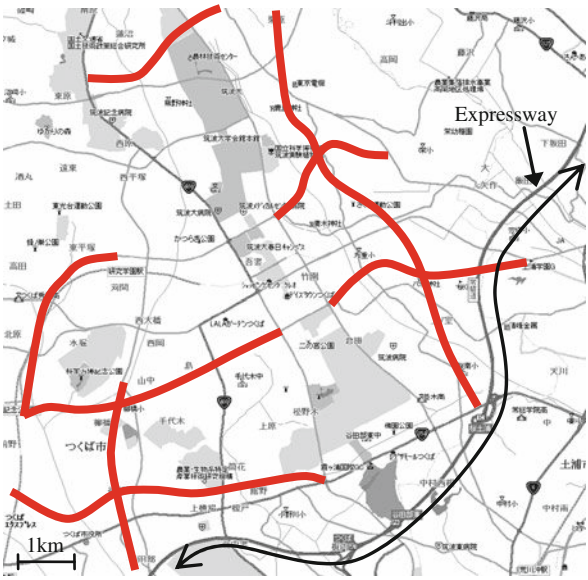


Fig. 3.28 Layout of the assumed new roads

(a) Before the opening of the new roads

(b) After the opening of the new roads

Fig. 3.29 Change of the noise map according to the opening of the new roads

from the traffic flow before and after the new roads. The noise increase along the new roads is clearly seen. However, in the other roads, the effect of the noise reduction is not clearly seen because the speed of the traffic flow was increased. This result indicates the difficulties of the noise reduction measure. The decrease of the CO_2 emission by the opening of the new roads was predicted to be 0.6% (from 43.1 to 42.9 ton CO_2/h). This is ascribed to the traffic volume concentrated in the city center being dispersed, thus the traffic flow generally becoming smoother.

3.8 Modeling Details of Advanced Case Studies

3.8.1 Capacity Reduction by a On-Street-Parked Vehicle

On-street-parked vehicles substantially reduce road capacity, especially in urban areas. However, it is quite difficult to completely remove all on-street-parked vehicles from roads. The feasible and effective parking management would be to control parking places, time of day and duration without serious damage on traffic flow.

AVENUE represents vehicle movement using Hybrid Block Density Method (HBDM) (Horiguchi et al., 1994). In HBDM, each lane is divided into blocks (block length depends on the free-flow speed and scan interval), and the traffic flow between neighbouring blocks are calculated by the fundamental diagram given for each blocks (capacity of block is explicitly defined). Therefore, to reproduce the effect of on-street-parked vehicles, capacity of the block is reduced.

Figure 3.30 shows the image of on-street-parked vehicle model and the example of reduction pattern for capacity of block. The on-street-parked vehicle model decides the reduction rate of capacity of blocks on lane 1 and lane 2 depending on the extra margin of the lane width. As shown in Fig. 3.31, the reduction pattern of the capacity is made by referring to the previous survey and research (Japan Society of Traffic Engineers 2000). However, the capacity reduction is considered at most only for two lanes from the shoulder side.

3.8.2 LRT and Public Transportation Priority System

In recent years, variety of public transport systems have been proposed and tried to combine for regional revitalization and protection of the environment. In traffic simulation work, we have two points of the evaluation for public transport. We have to first evaluate how much a new transport system influences on the current traffic condition and second how the system could be effectively operated.

Introduction of LRT sharing a roadway with vehicles could reduce traffic volume and CO_2 emission. On the other hand, the road capacity assigned to vehicles would also decrease because of the limited road space. Figure 3.32 shows an example of the LRT simulation with PTPS (Public Transportation Priority System) signal control,

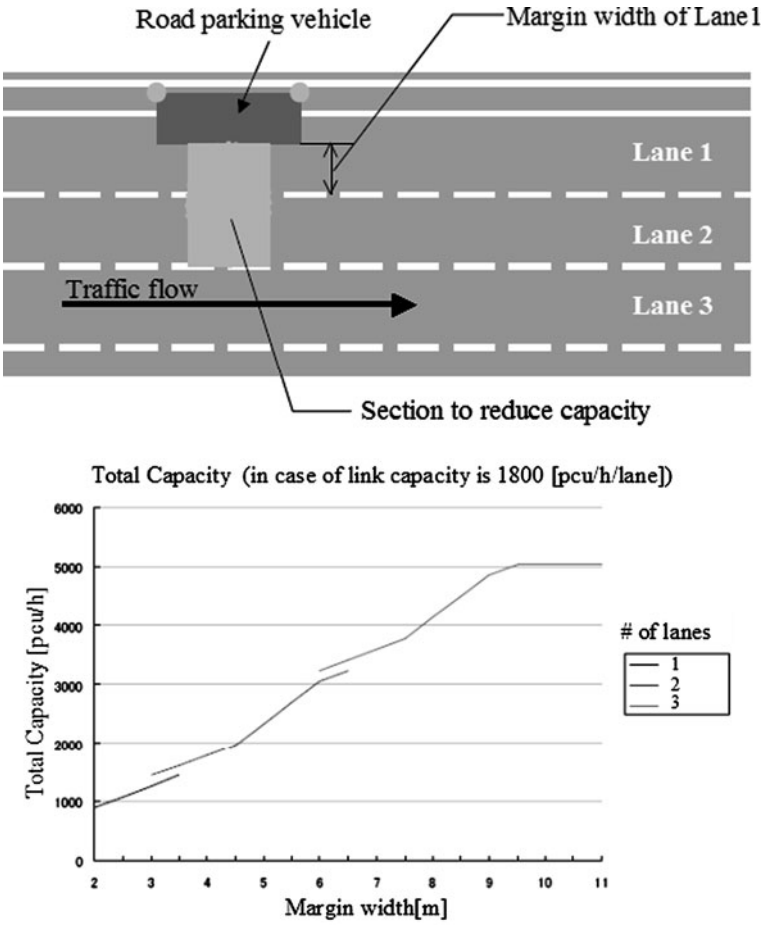


Fig. 3.30 Capacity reduction due to on-street parking

which prioritizes LRT to get through intersections without stopping. When LRT goes through a detector, a signal controller located at the downstream switches the traffic light so that the LRT do not have to stop.

3.8.3 Pedestrian Crossing

Several pedestrian simulation models have been proposed for various purposes. For a macro or mesosimulation model like SOUND (Yoshii and Kuwahara, 1995) or AVENUE, behavior of pedestrian is normally quite simplified and the impact of pedestrian behavior onto traffic is generally represented as the link capacity reduction. For instance, in AVENUE, the pedestrian behavior at an intersection results in

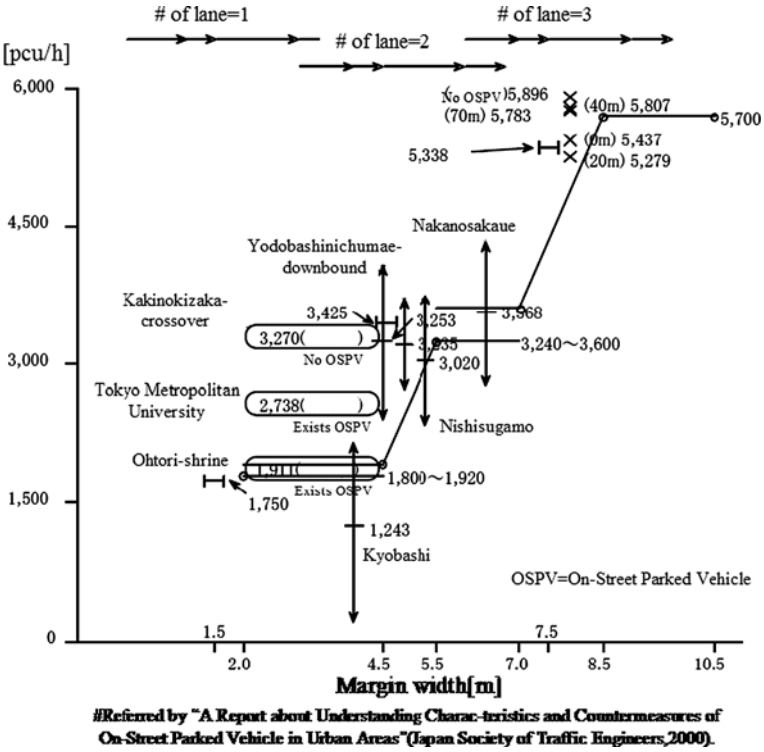


Fig. 3.31 Capacity in relation to the effective lane width in urban streets

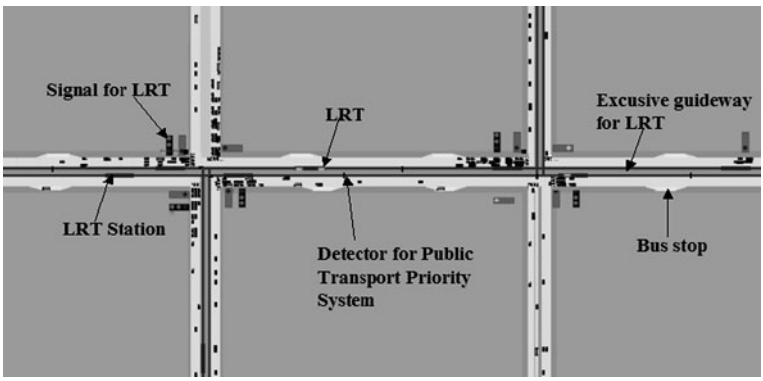


Fig. 3.32 An example of the LRT simulation with PTPS

the saturation flow rate reduction. Depending on the number of pedestrians crossing an intersection approach, the saturation flow rate is prepared in advance. The capacity or saturation flow rate reduction due to pedestrians has been measured in several fields in urban areas.

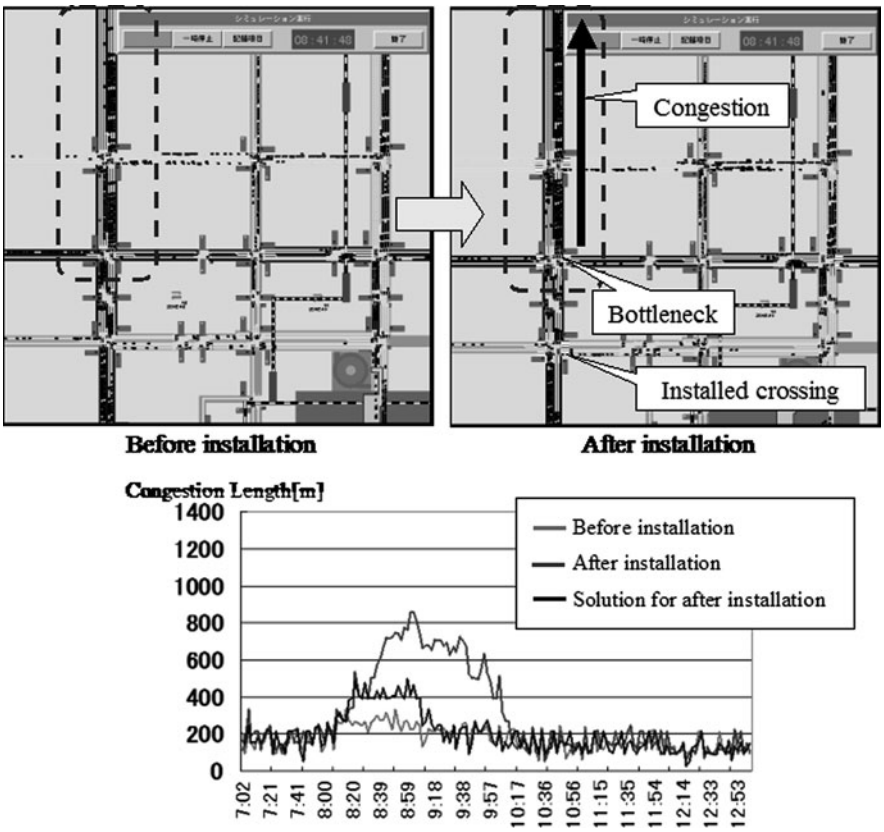


Fig. 3.33 Example of traffic simulation using crossing model

Figure 3.33 shows a case study, in which AVENUE evaluates traffic performances before and after installing a pedestrian crossing on a major arterial. As in the figure, the congestion length from a bottleneck after the installation, heavy congestion on the left-side street occurs due to the capacity reduction of right and left turns. Therefore, the signal timing is adjusted to mitigate the congestion as shown in the solution case.

References

Asano M, Nakajima A, Horiguchi R, Oneyama H, Kuwahara M (2003) Traffic signal control algorithm based on queuing model using ITS sensing technologies. In: Proceedings of 10th World Congress on Intelligent Transport Systems, Madrid

Furukawa M, Kuwahara M (2000) Automatic tuning of parameters in a network traffic simulation model. In: Proceedings of 7th World Congress on Intelligent Transport Systems, Turin

Hanabusa H, Yoshii T, Horiguchi R, Akahane H, Katakura M, Kuwahara M, Ozaki H, Oguchi T, Nishikawa I (2001) Construction of a data set for validation of traffic simulations. JSCE 688/IV-53:115–123

Hino T, Sato M, Uehara K, Imamura K, Akahane H, Haruta H, Horiguchi R, Sunahara H (2004) HAKONIWA – Application development environment for InternetCAR system. In: Proceedings of 11th World Congress on Intelligent Transport Systems, Nagoya

Horiguchi R, Katakura M, Akahane H, Kuwahara M (1994) A development of a traffic simulator for urban road networks: AVENUE. In: Vehicle Navigation and Information Systems Conference Proceedings, Yokohama, pp 245–250

Horiguchi R, Kuwahara M, Nishikawa I (1995) The model validation of traffic simulation system for urban road networks: 'AVENUE'. In: Proceedings of 2nd World Congress on Intelligent Transport Systems, Yokohama, vol IV, pp 1977–1982

Japanese Society of Traffic Engineers (1999) Standard verification process for traffic flow simulation model. http://www.jste.or.jp/sim/manuals/VfyManE.pdf. Accessed 25 Nov 1999

Japanese Society of Traffic Engineers (2000) Analysis on the capacity affected by roadside parking vehicles on urban streets (in Japanese)

Kono S, Oshino Y, Iwase T, Sone T, Tachibana H (2004) Road traffic noise prediction model 'ASJ RTN-Model 2003' proposed by the Acoustical Society of Japan – Part 2: Calculation model of sound power emission of road vehicles. In: Proceedings of ICA 2004, Kyoto, Japan

Kuwahara M, Oneyama H (2003) Automatic calibration of parameters in a traffic simulation model. In: Proceedings of 10th World Congress on Intelligent Transport Systems, Madrid

Oneyama H, Kuwahara M, Yoshii T (1996) Estimation of time dependent OD matrices from traffic counts. In: Proceedings of 3rd World Congress on Intelligent Transport Systems, Orlando

SAE (1982) Automotive engineering handbook, 2nd edn. The Society of Automotive Engineers of Japan, Japan (in Japanese)

Tsukui K, Hanabusa H, Oneyama H, Oshino Y, Kuwahara M (2009) CO_2 and noise evaluation model linked with traffic simulation for a citywide area. Int J ITS Res 7(1):59–65

Uehara K, Watanabe Y, Sunahara H, Nakamura O, Murai J (1998) InternetCAR – Internet connected automobiles. In: Proceedings of INET'98, Geneva, Switzerland

Yoshii T, Kuwahara M (1995) SOUND: A traffic simulation model for over-saturated traffic flow on urban expressways. In: Proceedings of 7th World Conference on Transportation Research, Sydney

Chapter 4
Traffic Simulation with Paramics

Pete Sykes

4.1 Introduction

SIAS is a transport-planning consultancy based in Edinburgh, Scotland, since 1974. SIAS has always been known for technological innovation through its early work in GIS and road mapping and in the analysis of road traffic. In 1978, SIAS's personnel produced NESA (Network Evaluation from Survey and Assignment). NESA is used to assess proposed road schemes and measure their impact on traffic and on the economy in the surrounding area. NESA consists of two submodels: a traffic assignment model, which allocates vehicles to the road network, and an economic assessment model, which calculates the user costs, benefits and economic return of the scheme.

Experience with NESA led SIAS to propose a more advanced form of transport modelling based on individual vehicle movements. This was intended to address the shortcomings inherent in modelling congestion through simple assignment and provide the ability to include the small-scale effects of individual vehicle behaviour in the assessment of a road scheme.

In 1986 SIAS wrote a prototype of an individual vehicle modeller and subsequently won European Commission's finance to develop it further in collaboration with various other institutions. These included the University of Namur in Belgium and Edinburgh University Parallel Computing Centre. In 1998, development of the software was taken back in house by SIAS and in 2000 the first commercial release of S-Paramics was made.

In 2006, PEARS (Program for Economic Assessment of Road Schemes) was released. PEARS takes the output of a microsimulation model and uses that to perform an economic assessment of the scheme adhering to UK Department for Transport guidelines.

P. Sykes (✉)
SIAS Ltd, 37 Manor Place, Edinburgh, Scotland, EH3 7EB
e-mail: pete.sykes@sias.com

J. Barceló (ed.), *Fundamentals of Traffic Simulation*,
International Series in Operations Research & Management Science 145,
DOI 10.1007/978-1-4419-6142-6_4,

4.2 Applications

S-Paramics is used on a wide variety of transport-modelling projects. At one end of the range, there are single-junction improvement assessments and small-scale traffic impact analyses. Larger scale models include wide area city or regional models used to assess the impact of major schemes or used as a continuously available planning tool to study the effect of a number of road traffic schemes in one area.

Interfaces from S-Paramics to urban traffic control systems are available which allow the simulation models to be used to develop ITS control strategies for incident and event management as well as to investigate options for optimisation of urban control systems and of motorway control systems.

The strength of S-Paramics is in its ability to apply microsimulation to large area models. The 'micro' in microsimulation reflects the level at which the interactions between vehicles are modelled, not a limit to the size of the geographical area in a model. Large models may cover tens of squares of kilometres with hundreds of zones and hundreds of kilometres of road network.

4.3 Model-Building Principles

4.3.1 Principles

S-Paramics has a general philosophy of requiring the modeller to create a model of the road network in which drivers move, with a single-minded goal of reaching their destination, as efficiently as possible while obeying the rules of the road and interacting safely with other vehicles in the simulation. S-Paramics avoids the use of modelling artefacts, which are impossible to implement on the actual road network, even though they may make the task of base model calibration easier.

For example it is not possible in S-Paramics to prescribe, at any point, the proportion of vehicles using each lane on a link. It is however possible to define where drivers become aware of a junction and hence where they will start to get into the correct lane for their manoeuvre. Using this mechanism the flow of vehicles in each lane occurs naturally rather than by being prescribed. If the model fails to replicate the observed data, then the modeller is encouraged to look at the assignment of vehicles moving through this junction and where they become aware of the junction, and hence their lane choice.

By requiring this descriptive method of construction and calibration, a model will have a more robust predictive ability as changes to travel demands and network topology are tested.

4.3.2 Network Construction

Building a simulation model compromises a number of key stages. Firstly the scope of the area to model for the study is established and the road network and

junctions are created to cover this area. Next, a zoning scheme is created and the traffic demands between these zones are entered in demand matrices. This also requires that the proportions of different vehicle types in these demand matrices are described. Passenger transport and time-dependent properties of the network must be included in the model and any graphics such as annotation, aerial photos and 3D objects added.

Calibration is the process of refining the network such that the interactions of the vehicles in the modelled network operate as they do in reality on that road network.

4.3.2.1 Road Network

S-Paramics describes roads as a set of pairs of one-way *links* joined together at *nodes*. Links represent the roads in the network and thus have attributes such as speed, width, number of lanes and any lane-based restrictions, e.g. bus lanes. A node typically marks an area where links join to form junctions. Nodes can also mark points where links change characteristics, which can be any of their attributes, such as speed limit, number of lanes or a change in curvature or direction (Fig. 4.1).

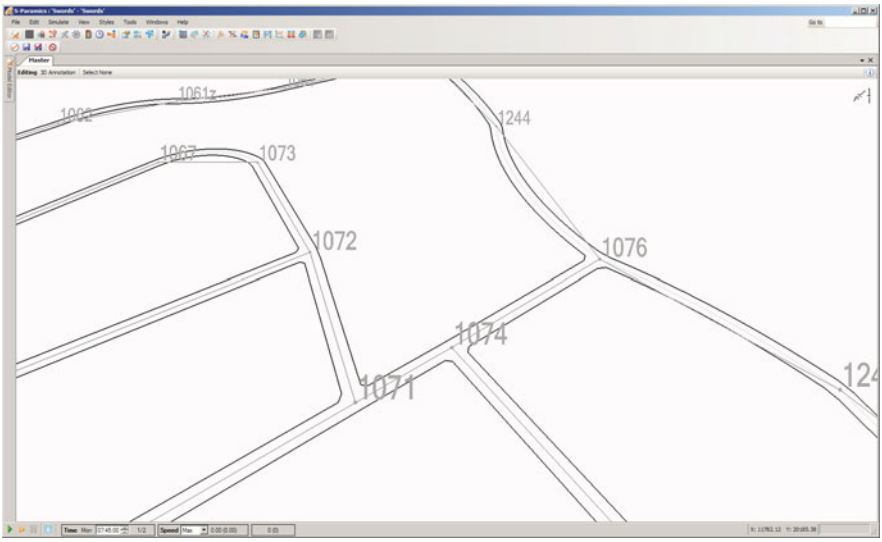


Fig. 4.1 Links and nodes

At junctions, stop lines mark the points where vehicles wait for gaps in oncoming traffic and where they move from one link to another. The stop lines also serve as locus points to describe the paths that the vehicles will follow.

S-Paramics uses a hierarchy of major and minor links to control the routeing of vehicles through the road network. The major links are, in effect, the sign-posted routes and the minor links are the secondary network. Drivers are classified

as ‘familiar’ and ‘unfamiliar’ with the latter perceiving the minor routes as less attractive than the major routes.

To manage the task of creating links, S-Paramics has a set of link categories. The concept is similar to that of a word processor’s style sheet. Categories can be defined and applied as the model is built and a set of pre-defined categories is also available. Links can later be modified either by category or individually.

4.3.2.2 Signals

Signals are added to create signalised junctions in the model. This is a three-stage process.

1. Specify the junction. It may be a single node or, if the junction geometry is more complex, a group of nodes.
2. Add the signal phases, linked to vehicle movements.
3. Add a stage plan to determine the times at which phases are active. If detailed phase offsets and specific phase inter-greens are required, these may be included.

4.3.2.3 Zoning Scheme

Zones represent the network entry and exit points for vehicles. Zones may either be ‘network connectors’, in effect the edges of the network, or they may represent land use areas, possibly already described in a strategic planning model.

Figure 4.2 shows both types of zones. Zones 2781 and 090 are area zones representing post code areas, city blocks or land use areas. The traffic originating or terminating in these zones has a direct relationship with the activities carried out in these areas. Zones 9020 and 8005 are connections to areas outside the model.

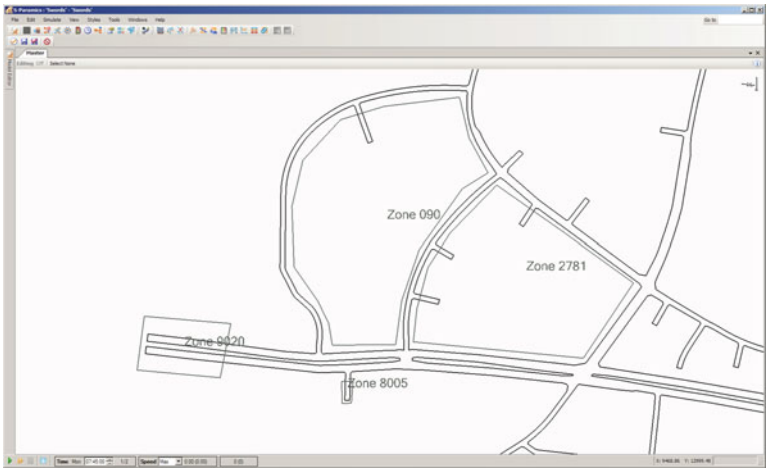


Fig. 4.2 Zones

Simple, single-junction or corridor models may require only network connector zones; larger models which cover wide areas will also require area-based zones.

4.3.3 Vehicles and Demand

4.3.3.1 Vehicles

S-Paramics can model many vehicle types ranging from cars to HGVs to light or heavy rail. Vehicle types may be used to differentiate between

- vehicles of different types, e.g. cars, LGVs, HGVs and taxis
- vehicles with different physical characteristics, e.g. small, medium or large cars
- vehicles with different journey purposes, e.g. commuting, business, leisure
- vehicles with different emission characteristics
- passenger transport vehicles

Each vehicle has a set of basic physical properties such as size and number of sections (for articulated vehicles), maximum speed, acceleration and deceleration. Other parameters are also defined including the demand matrix it will use and the engine type, which governs the quantities of emissions the vehicle will generate.

Vehicles in the simulation have a notional driver who is ascribed two characteristic levels: *awareness* and *aggression*. Awareness controls the likelihood that a driver will collaborate with others on the road, e.g., by adjusting headways to allow others to make lane changes. Aggression controls how a driver behaves with respect to speed selection and lane use. A more aggressive driver will tend to travel faster and delay lane changes required for a turn, preferring to use lanes with faster moving traffic.

Vehicle types also define a physical appearance. While this has no effect on the results of the simulation, it has a significant impact when presenting the model to a wider audience – perhaps at a public enquiry. Presentation options range from complex, realistic shapes to simple cuboids. The decision about which to use depends on the target audience and the speed of visual rendering required.

4.3.3.2 Demand

S-Paramics uses origin–destination (OD) demand matrices to control the loading of vehicles into the network.

OD matrices may be disaggregated into a number of *matrix levels*. These matrix levels can represent different vehicle types, different journey types or they can add development-specific demands to an existing model.

For example, a model may have a base matrix level corresponding to the current demand and a second matrix level which contains demands relating to a new development. Similarly a model may have a base matrix level with the car-based demand and a second matrix level with HGV-based demand. The departure time profiles for

the two matrices may be adjusted separately. This facility was used in one case to model the departure time peaks of HGV traffic at a busy freight-oriented shipping port.

Once the zones, vehicle types and matrix levels have been defined, the OD demand can be included in the model. This is a set of tables of OD demand, one for each matrix level.

4.3.3.3 Profiles

OD demand may be given a detailed departure time profile, which varies demand over time as the simulation model runs.

Consider the example in Fig. 4.3. The hourly demand for each hour is the same, only the profile of the demand changes. The 1-h profile overestimates the number of vehicles in the initial half hour of the simulation and underestimates the peak in the middle of the 3-h period. The 1-h profile would not allow the model to accurately reproduce the traffic conditions in either the peaks or the troughs.

S-Paramics therefore allows the modeller to adjust the number of vehicles released in each 5 min interval to accurately reproduce the prevailing traffic flows.

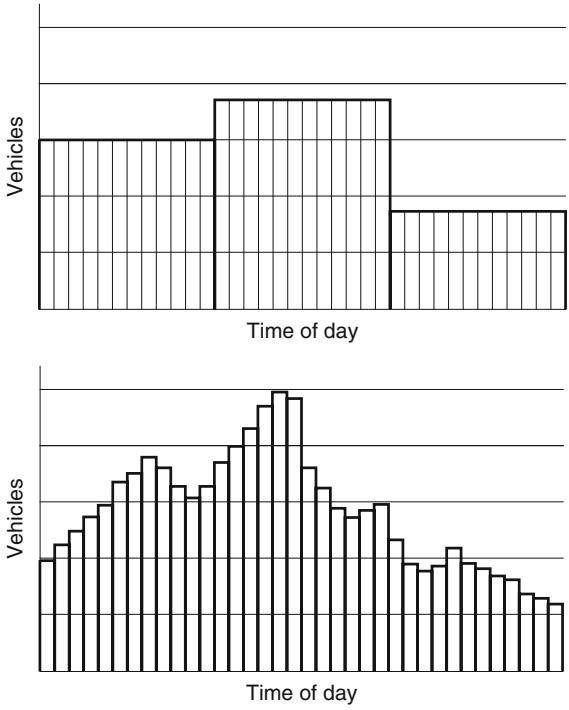


Fig. 4.3 Demand profile detail: 1-h demand vs. 5-min profiled demand

4.3.3.4 Passenger Transport

Passenger transport vehicles are programmed into the simulation by specifying a set of fixed routes, a set of bus stops or stations along the route and a schedule of release times for each passenger transport service.

Bus routes are created automatically along the shortest path between specified start and end links. Further intermediate links can be specified if the required route does not follow the shortest path. When the model is subsequently updated to simulate a design scenario, its bus routes are automatically re-created with the same start, intermediate and end links and will automatically include any road network changes.

4.3.3.5 Time Periods

Time periods are used to segment the overall period for which the model runs. Typically a period boundary marks where a change occurs in the model. These changes could be the following:

- lane restrictions, e.g. bus-only lanes change to become traffic lanes outside the peak period
- junction priorities change, e.g. a turn is barred in a peak period
- vehicle-type proportions, e.g. ratio of HGVs to cars

Time periods may also be used to implement changes in demand or changes in signal timings but often it is better to implement these changes with signal time plans or with the demand-profiling facility.

4.3.4 Presentation

Annotation, aerial photographs, buildings and other 3D shapes may be incorporated into the model to enhance its presentation quality. The amount of effort to invest depends on the use and intended audience for the model and the budget allocation for presentation.

The quality can vary from a simple schematic network representation to a full 3D virtual reality model. Simpler background graphics can use the appropriate generic buildings, landscape objects and street furniture that are available with a standard S-Paramics installation. A high-end presentation may require development of a 3D landscape or cityscape of the modelled area. Figure 4.4 shows three images of the same simulation scene with different presentation levels. Figure 4.5 shows S-Paramics simulation data included in a complex 3D model.

Fig. 4.4 Presentation options

Fig. 4.5 3D presentation (image courtesy of baseplusworld and Truescape)

4.4 Simulation Model

There are two components required when simulating vehicle movements and driver behaviour. The first is the provision of the environment for vehicles to move through; the second is the implementation of the behavioural algorithms that simulate driver responses to their perceived environment. Both are of equal importance in the implementation of a microsimulation model.

4.4.1 Environment

4.4.1.1 Trajectories and Geometry

The nodes, links and stop lines that are created by editing the skeletal S-Paramics model are processed as the model is read to convert them into a set of lane-based trajectories using the stop lines as locus points. These trajectories represent the mid-points of the lanes and are the default paths for vehicles as they traverse links and junctions. They are derived from straight lines or arcs on simple link sections and Bezier curves, which provide smooth transitions joining the lines and arcs together, where vehicles travel across nodes.

Trajectories are interrupted by wait points, placed at the stop line positions, where vehicles are required to stop or give way. This may be at traffic signals, or to wait for

a suitable gap in other streams of vehicles. These wait points will normally coincide with the physical markings on the road where vehicles come to a halt.

A driver will 'see' along the trajectory lines with which its own planned trajectory either crosses or merges. This enables it to identify those vehicles with which it is in contention for space on the road. The driver is therefore able to calculate the relative times to the crossover or merge point and either wait or adjust the speed of the vehicle to merge into the gap accordingly (Fig. 4.6).

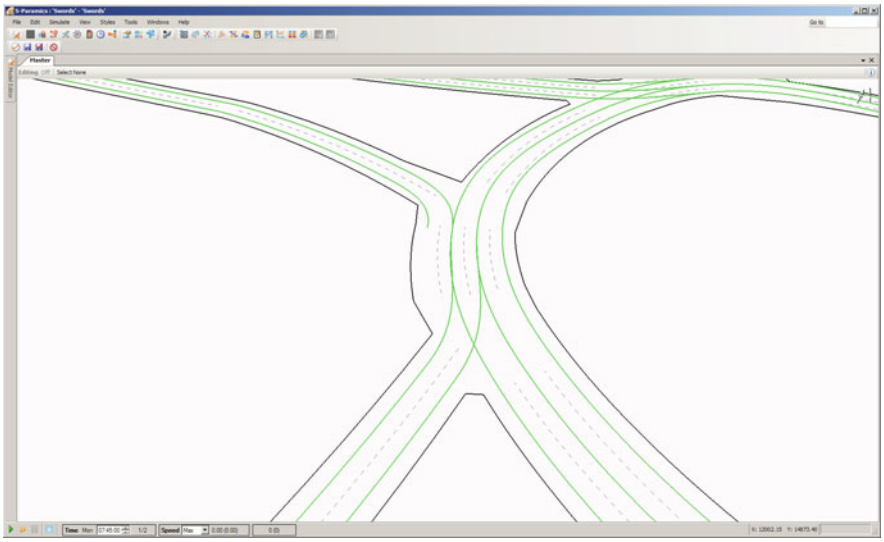


Fig. 4.6 Trajectory lines

The trajectories identify the lanes in the road network and give vehicles a path to follow. This applies to the simple case where no lane changing takes place. In reality, as drivers react to different flows in lanes or have to get in the right lane for a manoeuvre, they must change lane.

When a lane change is required and a suitable gap in the adjacent lane has been found, the vehicle initiates a lane change. To accomplish it, a smooth intermediate trajectory is generated and the vehicle moves on that to transfer from one lane to the other. This intermediate trajectory is temporary. It is generated as the vehicle starts its manoeuvre and is deleted once it has finished (Fig. 4.7).

A vehicle with a large target speed differential between it and the vehicle ahead may initiate an overtaking manoeuvre. This requires pulling out onto the other side of the carriageway against the opposing traffic flow. It will first assess the road for visibility and if the opposing carriageway is both visible and clear of oncoming traffic for sufficient space, then a new trajectory will be generated to allow it to pass.

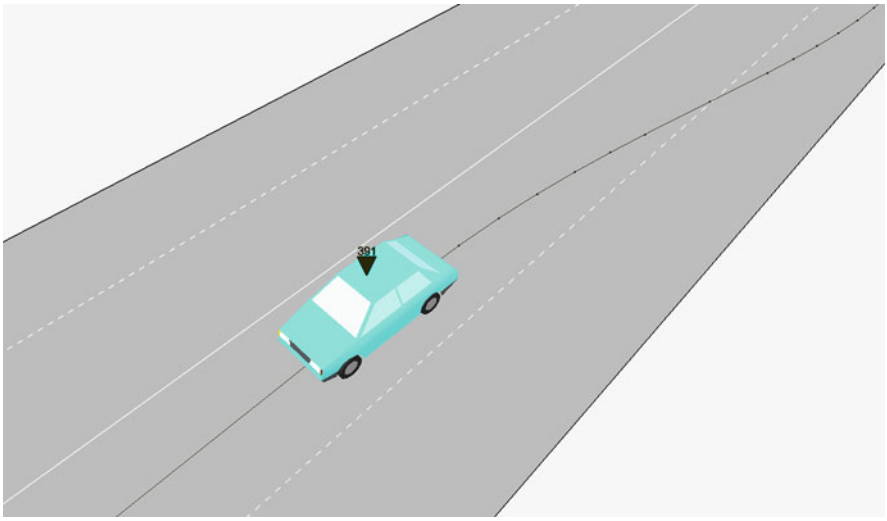


Fig. 4.7 Lane change intermediate trajectory

4.4.1.2 Hazards

Away from the influence of junctions and changes in road layout, a vehicle will set its speed and lane choice by its interactions with other vehicles and the road layout. This free choice area is typical of long sections of road with no immediate junctions that require the driver's attention.

Closer to a junction, or closer to a node where the road layout changes (e.g. a lane gain or a lane drop), a driver must react to the imminent change. The driver will reconsider which lanes may be used for the forthcoming manoeuvre and whether a lane change is required. A location where any such action is required is referred to as a 'hazard' and the location where drivers become aware of a hazard on the network is referred to as the 'signpost'. This models the reality of, for example, an off-ramp on a highway with a signpost positioned some distance before the junction to alert drivers to its presence.

Figure 4.8 shows a 'diverge' hazard at the off-ramp of a grade-separated roundabout.

Drivers in the simulation are given attributes of aggression and awareness which control their responses as they set their speed and headway and as they react to signposts warning them of imminent hazards. The signpost is the first place at which drivers become aware of the hazard; their reaction to it will be at some point between the signpost and the corresponding hazard. For example a driver in the free choice region of the road with no active signposts or hazards will choose which lane to use from the list of available lanes and by deducing which lane offers the best progress. More aggressive drivers will select lanes with faster traffic based on smaller differentials in vehicle speed. Less aggressive drivers will tend to stay in lane and change

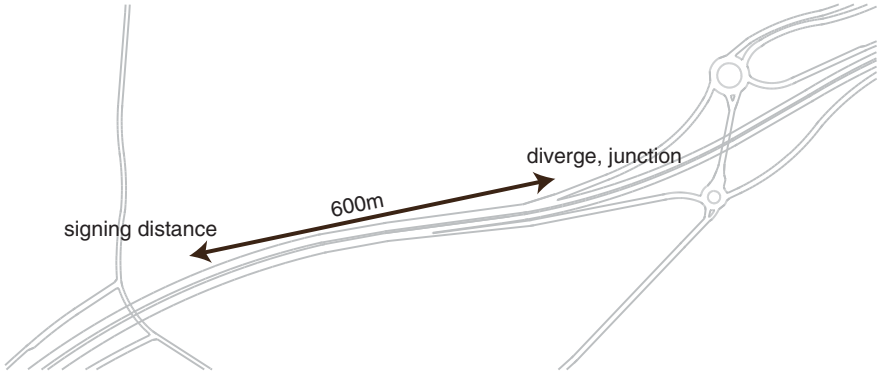


Fig. 4.8 Diverge hazard

only when the speed differential is high. On passing a signpost, the less aggressive drivers will re-evaluate their lane choice and if their route determines that they must change lane to make the manoeuvre, they will attempt to do so. More aggressive drivers will re-evaluate their lane choice later and continue to select the best lane for forward progress until that point.

The combination of vehicle trajectories with associated wait points and the location of the hazards and signposts are the key components of the representation of the road infrastructure.

4.4.1.3 Vehicle Behaviour

During each time step, each vehicle performs the following actions:

1. Observes its surrounding environment and evaluates its current options for gap acceptance; speed; acceleration; lane choice; lane change; and trajectory propagation.
2. Selects a target lane, trajectory and acceleration from the options it calculated in stage 1 and updates its location, speed etc.
3. Generates model outputs (statistics) for calibration or for comparison with the base or with a design model.

The calculations in stage 1 determine the vehicle's actions and are composed of a set of discrete behaviour facets which combine to define the vehicle's overall behaviour.

The behaviour facets are controlled by a set of low-level simulation behaviour configuration files which govern the facets in use and define any configuration parameters. These configuration files also prioritise the behaviours such that if one returns an overriding result, the remaining ones will not be evaluated. Introducing a

new behaviour can be achieved by writing a behaviour module and including it in the decision tree in the configuration files.

4.4.1.4 Lane Choice and Lane Change

S-Paramics implements a structured hierarchy of lane choice and lane change decisions through a system of discrete lane ranger and lane suggester behaviour facets that are combined to derive a final lane choice. The lane rangers each provide a range of lanes that the vehicle can use to follow its route to its destination. The lane suggesters then supply weights for each lane in this range. Finally the lane chooser will select the lane with the highest weighting.

If a need for a lane change is identified through this process and that need has been consistently identified for a prescribed time, then the vehicle will start searching for a suitable gap in adjacent lanes. It will then adjust its speed to position itself to make the change through the 'lane change' acceleration suggester. Finally, when the adjacent gap is acceptable, it will generate a temporary trajectory to bridge the gap from one lane to the next.

Lane rangers operate in two areas: those that take account of hazards and those that act independently. There are just three lane rangers operating independently of hazards (Table 4.1).

The hazard-based lane rangers deal with the task of refining the range of lanes as a vehicle approaches a change in the road layout or as it approaches a junction. These rangers include the facets listed in Table 4.2.

If, at any point in the process, the lane range has been reduced to a choice of just one lane, no further evaluation is made.

Table 4.1 Lane rangers

Lane restriction	Returns the range available to a vehicle after taking into account any lane-based restrictions
Bus lane	Similar to the lane restriction ranger but applicable to passenger transport vehicles as they approach their next stop
Stay in lane	Restricts the vehicle to its current lane choice if lane changing is prohibited in this region

Table 4.2 Hazard-based lane rangers

Lane closure	Takes into account any approaching lane closures
Road confluence	Prepares a vehicle to give way to merging traffic by increasing its lane range to include the offside lane
Road diverge	Selects the lane range to those which continue directly into the vehicle's chosen path and the diverge
Junction	A class of lane rangers which select the lane for any approaching turn based on the lanes made available for that manoeuvre

Following the lane range selection and if there is a choice of lanes, the lane suggesters then weight each of the lanes in the range. The suggester facets are given in Table 4.3.

Table 4.3 Lane suggesters

Congestion	Slow-moving vehicles with a high aggression value that are caught in congestion will try and move towards neighbouring lanes. Vehicles will move towards the lane with the greater speed difference
Drop	Give an increasing weight to a lane change if there is a lane drop ahead
Accelerating	Vehicles tend to stay in lane if they are accelerating at a rate above a specified threshold. This suggester will give weight to the current lane
Behind bus	Attempt to pass a bus and avoid the nearside lane if there is a bus in it
Gradient	HGVs tend towards nearside lane when travelling up inclines
Nearside	Vehicles tend towards the nearside lane (UK traffic)
Lane spread	Spread lane use across all lanes based on lane occupancy
Vehicle behind	Aware vehicles will move to nearside lanes when a faster vehicle approaches from behind
Avoid incident	Avoids lanes containing an incident
Passing	Allows passing by changing lane on dual carriageways
HOV lane	Move to an HOV lane if it is permitted and advantageous. This may require a change of more than one lane

4.4.1.5 Speed and Acceleration

In each time step, each driver consults the series of headway and acceleration suggester facets defined in the configuration file for the vehicle. In addition, a vehicle's maximum speed and acceleration (or deceleration) can be limited, by corresponding modifier facets, to suit its current circumstances, e.g. a reduced maximum speed for HGVs travelling uphill. This combination of behaviour facets allows the driver to evaluate the acceleration it requires for its impending manoeuvres and the headway required for vehicle following.

The speed and acceleration modifiers are applied in each time step before calculating accelerations or headways, the results of which depend on the maximum values thus defined. The modifiers are listed in Table 4.4.

Table 4.4 Speed and acceleration modifiers

Drag and inertia	Reduces a vehicle's maximum acceleration depending on the values of its drag and inertia
Gradient deceleration	A simple model to modify deceleration according to gradient. It uses one of a set of modifiers for different types and weights of vehicles
HGV gradient deceleration	The HGV gradient speed and acceleration model uses a set of empirically derived factors to define a power curve that limits both maximum speed and maximum acceleration on uphill inclines

Table 4.5 Headway suggesters

Default	Sets up an initial headway for each vehicle using a linear spread based on the driver's aggression and awareness around the mean headway. This will be modified on links where the modeller has adjusted the mean headway. In addition, headway will be adjusted for HGVs
Signals	Vehicles approaching signals will have their headway modified depending on the turn they will make as they pass through the signals
Narrow	As a vehicle approaches a lane reduction and is within its signposting distance, the headway factor is adjusted so that a smaller headway is accepted as the lane change becomes more urgent

The headway suggesters are given in Table 4.5.

The list of acceleration suggesters is given in Table 4.6.

Each of these suggesters returns either 'no suggestion' if it is not applicable to the vehicle's environment or an appropriate acceleration or deceleration value.

The acceleration arbiter then selects the value to use, capped to the physical limitations of the vehicle. The arbiter is quite simple in that it selects the lowest acceleration value suggested, assuming that the lowest acceleration determined by the most restrictive speed constraint will be the one used by the driver.

4.5 Gap Acceptance

When the forward projection of a vehicle trajectory crosses or merges with another trajectory, then a contention is generated. The contention handler will use the junction priority rules to determine which vehicle has the higher priority. Vehicles with 'major' priority will not change their speed, while those with lower priority will slow to a halt at the corresponding stop line. At that stop line, or at the prescribed 'visibility' distance ahead of the stop line, a driver will look for vehicles on the contending trajectories and assess what acceleration is required to cross or merge. In making the crossing or merging manoeuvre, a driver will allow for a time-based headway as a safety margin between two vehicles.

Vehicles are held at their stop lines until a gap in traffic on contending trajectories is found which allows the vehicle to accelerate to reach the contention point with adequate headway between it and the contending vehicle.

In congested circumstances, drivers are observed to collaborate with each other to allow vehicles to cross congested slow-moving or stationary streams of traffic or merge with them. These forcing through manoeuvres break the rules regarding headway and merging but in practice they occur. Within the simulation, these small enabling movements are not reproduced. Instead 'force cross' or 'force merge' is allowed to let vehicles break the normal headway rules and cross or merge after a waiting time has passed.

Table 4.6 Acceleration suggesters

Bus acceleration	Sets the bus acceleration to come to a halt at its next stop and to remain at its stop for the required bus top dwell time duration
Clear exit	Provides compliance with yellow box junction rules
Curvature	Provides an acceleration to set a safe speed for a curved trajectory
End speed	Provides an acceleration to bring the vehicle to the link end speed, if one is supplied for the link
Hazard	Brings the vehicle to a halt at the wait point of a junction
Following	Converges on the desired headway for this vehicle
Lane change	Provides an acceleration for a current lane change manoeuvre
Lane end	Prevents vehicles from driving off the end of the lane in a ramp merge
Let in	Provides an acceleration which will allow other vehicles into this vehicle's lane. This takes into account vehicles ahead in adjacent lanes and their relative speeds and positions
Lane mapping	Provides the acceleration to apply in a zip merge. The vehicle will adjust its speed to give way to another vehicle and find a suitably sized gap in a lane adjacent to itself
Overtake (opposite carriageway)	Provides a deceleration for an overtaking vehicle when there is a vehicle approaching in the opposite direction
Outside friction	Constrains a vehicle's speed as it passes vehicles in adjacent lanes
Ramp merge	Provides an acceleration to merge with main line traffic from a ramp. This may be achieved either by accelerating to achieve an adequate headway in front of an adjacent vehicle on the main carriage way or decelerating to merge behind it. If the distance to the head of the ramp is small, the required headway is reduced
Speed limit	Provides an acceleration based on a vehicle's target speed for a link. This speed is set by the advisory speed limit on the link and the vehicle's maximum speed. The target speed may exceed the posted speed limit for the link for drivers with high aggression
Undertaking	Either constrains a vehicle to less than the speed of the vehicle in the outside lane or, if US style freeway behaviour is permitted, allows the vehicle to pass another in any lane
Want lane change	Sets an acceleration to give a suitable headway between this vehicle and one in its target lane in preparation for a lane change.
Contention	Provides acceleration values for vehicles when their paths overlap at junctions
Converge	Provides an acceleration to stop at the give way line, or to accelerate through or away from the give way line if there is no contention. With other vehicles
Roundabout Approach	Roundabout is a specific contention handler for roundabout approaches. Like 'Converge', it provides either an acceleration to stop at the give way line, or one to accelerate through or away from the give way line.

4.6 Assignment

S-Paramics has a range of algorithms to find routes for vehicles within a simulation model. The most simple is an 'all or nothing' assignment, applicable where there is no route choice in the model. The most complex routeing algorithms in S-Paramics segment the journey into sections between major waypoints and include the effects of a driver's imprecise perception of the relative costs of different routes.

The derivation of costs depends on a hierarchy of road classes and can include a process of learning both about congestion and a variable response to it.

4.6.1 Driver Knowledge

Drivers are classified as ‘familiar’ or ‘unfamiliar’ in the routeing system. This corresponds with their knowledge of the conditions of the network and how they perceive journey time costs on ‘major’ and ‘minor’ routes. Familiar drivers use secondary roads (the minor ones), while the unfamiliar drivers tend to keep to the main roads. The ratio of familiar to unfamiliar drivers can be set for each vehicle type. For example, it could be assumed that taxi drivers are 100% familiar. The driver attributes of aggression and awareness are also used to determine how they react to congestion, whether they accept the extra delay or whether they opt to save as much time as possible by using every available rat run.

4.6.2 Road Network

An S-Paramics road network is built with a two-level hierarchy of links (Fig. 4.9). The ‘major’ links correspond to the main roads, often characterised as the signposted routes. The ‘minor’ roads correspond to the secondary routes. The incremental cost of using minor routes is doubled (by default) for those drivers marked as ‘unfamiliar’.

At each junction there is at least one pair of tables of costs from each of the exits of the junction to each of the destination zones in the network. Each pair of tables

Fig. 4.9 Road hierarchy

comprises one table of costs for the familiar, and another for the unfamiliar, drivers. Vehicles approaching a junction will consult these tables and select the appropriate turn depending on the subsequent cost to the destination.

If there are link-based restrictions in the model or different vehicle types with different route cost coefficients, then multiple sets of route tables are generated at each junction. Generating multiple sets permits S-Paramics to accommodate physical, or statutory, restrictions in the model. This could include preventing tall vehicles from going under low bridges or restricting HGVs from city centre streets during peak periods.

4.6.2.1 Car Parks

S-Paramics has the capability to refine the start and end points of trips through the use of car parks. Car parks can be coded to reflect actual car parks or can simply be used to define multiple trip origin/destination points within an area. The car parks are associated with a single zone or multiple zones to connect the car parks with the origin/destination data in the corresponding demand matrices. Walking times are also assigned between each car park and the centre of any associated zones and contribute towards the overall trip cost. The car park which gives the lowest overall trip cost will be selected when deciding which car park to use to either begin or end the journey.

Car park occupancy can be constantly monitored during the course of the simulation. When a car park is full, vehicles can either queue at the entrance until spaces become available or re-route to alternative car parks with spare capacity. ITS systems may change the destination car park for individual vehicles.

4.6.3 Static Assignment

The generalised cost for a route is based on distance, predicted time and tolls.

$$\text{Generalised cost} = (A \times \text{time}) + (B \times \text{distance}) + (C \times \text{tolls})$$

The default values are {1,0.7,0}, which implies that the default cost is made up of approximately 59% of the predicted time for a trip, with the remaining 41% for the distance travelled. Variation of this cost may then be introduced by vehicle type and by driver perception. Each type of vehicle may have a different set of cost coefficients. One driver may interpret cost as equivalent to time, while another may minimise distance.

In evaluating the cost of a route, the incremental cost of ‘minor’ links is doubled (by default) for the ‘unfamiliar’ drivers in the simulation. Hence, when there is route choice, they are more likely to select the routes made up of ‘major’ links and less likely to use the ‘minor’ links or rat runs.

Finally, a driver’s perception of time or distance is not accurate and different drivers will tend to use a number of routes between two points. To include this

effect in the simulation, when a driver is selecting which way to turn at a junction, S-Paramics adjusts the calculated costs of the route to the destination from each exit by a random amount. The driver will then select the route with the lowest adjusted cost. In Fig. 4.10 two routes, highlighted in blue, are shown to a destination. One route may have a journey time of 10 min, while the other may have 10 min and 10 s. With 10% perturbation the relative costs overlap and both routes are used by drivers at this junction.

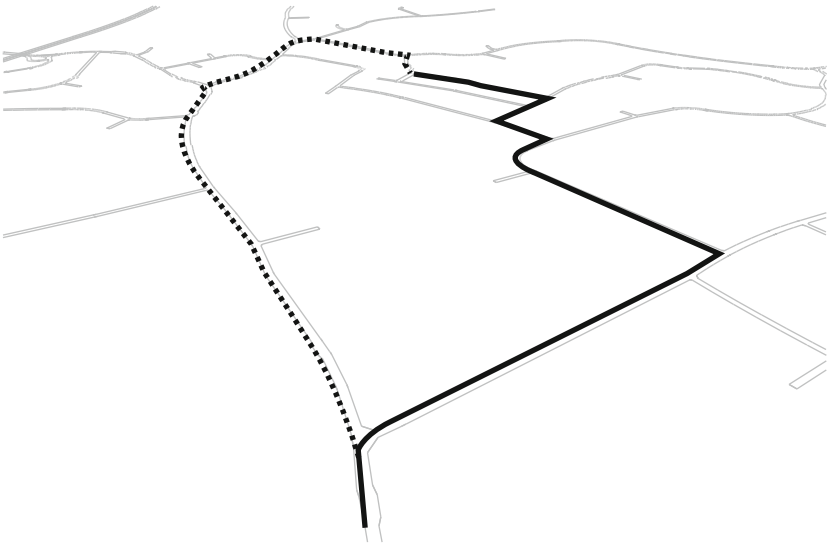


Fig. 4.10 Route choice due to perturbation

The selection process for the next turn based on route cost is repeated at every junction. Therefore, with route cost perturbation, as the vehicle proceeds to its destination, a range of routes will be used if the cost ranges overlap. The lowest cost route will tend to be preferred, although other plausible routes will be used too.

4.6.4 Dynamic Assignment

Dynamic assignment refers to the process by which drivers learn about the congestion they will encounter on a route to their destination. Then, in reaction to this acquired knowledge, their route is amended in an effort to minimise the perceived cost of the journey. As each individual reacts, the location and severity of the congestion will vary. Over time drivers will tend to avoid congestion and learn the quickest routes to their destinations, collectively minimising the aggregated journey costs.

Claims of uniqueness of a solution or for perfect optimisation are not necessarily meaningful in the stochastically noisy environment of driver behaviour, variable demand and imperfect perception of true trip costs. Hence a pragmatic approach to dynamic assignment is taken in S-Paramics.

Dynamic feedback within S-Paramics updates the transit times for each link at regular intervals throughout the simulation with the time achieved by the vehicles using that link. 'Familiar' drivers then use this actual cost instead of the basic cost which is derived from the link length and the advisory speed limit. In this way the 'familiar' drivers learn about the congestion on the road network and, in minimising their own journey costs, the overall flows are assigned optimally around the network. The process is a proxy for driver learning through repeated route testing and refinement.

Only those drivers marked as 'familiar' will be aware of the distribution of congestion on the network, while those marked as 'unfamiliar' will simply continue to form routes as before with their perception of journey time based on the advisory speed limits only.

Drivers that do receive feedback of link-based journey times may vary in their response to it. The most aggressive and aware drivers will take every opportunity to optimise their journey time, whilst the less aggressive and aware drivers will react only when more significant gains are to be made. S-Paramics allows modellers to vary the driver's response to knowledge of the network congestion based on the distribution of driver aggression and awareness.

4.6.4.1 Multiple-Level Routeing

As any driver who has asked for directions will be aware, the form of the response will tend to vary according to the distance involved. If the destination is nearby, then detailed instructions may well be given, e.g. 'First right, second left . . .'. If the destination is further away, then the instructions will tend to be based on key points on the journey, e.g. 'To the bypass, then to the New Town Junction . . .', rather than a long list of individual turns. Typically, the detailed directions are given only for the immediate segment of this overall journey.

S-Paramics embodies these two modes of route understanding by using 'waypoints' at strategic locations in the model. Waypoints are key route decision points, large junctions or identifiable areas. They are the points used to give high-level instructions for a long route. These macro-level routes between waypoints are subject to the same cost perception, cost perturbation and dynamic feedback as the micro-level routeing network, albeit with different values given for perturbation levels and feedback intervals.

A vehicle in a model using waypoints will have a 'macro'-level route linking its origin zone to its destination zone. This may pass through a number of waypoints. The overall journey is therefore segmented and each smaller segment is then traversed by vehicles using the 'micro'-level routeing system.

The effect is that the route choice between a zone and a waypoint or between waypoints is the same regardless of whether the trip is just that segment or a longer

trip which includes that segment. This condition is not necessarily satisfied when the basic algorithms of feedback and perturbation are used.

4.7 Calibration and Validation

Calibration is defined as the process of adjusting the parameters used in the model to ensure that it accurately reflects the input data. Validation is defined as the process of running an independent check on the calibrated model.

Two sets of observed data should be available during the model development process. One is used to calibrate the model, adjusting the parameters such that the modelled output adequately matches this observed data. The second set is used to verify that other aspects of the performance of the calibrated model are in agreement with this set of observed data.

In effect the calibration set of data is used by the modeller to adjust the model such that its behaviour matches the observed performance. Using the validation set for a second set of observations ensures that the adjustments made with respect to the calibration data have a positive effect on the overall level of agreement and not just on one particular aspect of calibration.

4.7.1 Assignment Calibration

If a model is a simple corridor or a single junction, then assignment is achieved with a simple all or nothing approach. If the model has route choice, then correct assignment of vehicles to the model is a key aspect to calibration. The assignment is calibrated by adjusting the following:

- demand matrices
- assignment, including
  - hierarchy of major and minor roads
  - junction modelling to derive realistic congestion costs
  - link-based cost modifiers

4.7.1.1 Demand

Accurate demand matrices are essential to achieve good calibration of assignment. Good data are essential to the creation of accurate demand matrices and this is typically taken from survey data, ANPR data and from census data and land use models.

S-Paramics includes a matrix estimation mode to update matrices so that they represent observed traffic conditions. This is an iterative process using the current assignment in the model and a prior matrix. It adjusts the demand matrix such that

the surveyed flows match the predicted flows. The matrix estimation process is controlled by constraints on changes to the matrix such that the numbers of vehicles flowing in to and out of zones are consistent with the size and demographics of the zone.

Traffic demand is rarely, if ever, constant for long periods of time. Flows build and decay during the day and the level of congestion is heavily dependent on the flow profile. S-Paramics is able to include time-based demand profiles at 5 min intervals and if data are available to support it, they should be used to control the release of vehicles into the simulation. Accurate profiled release is a key factor in model calibration to reproduce flows over short intervals and hence to realistically model queueing caused by short-term junction over-saturation which in turn affects dynamic assignment.

4.7.1.2 Assignment

Assignment principles in microsimulation models are similar to those in deterministic route choice modelling. Route choice is based on:

- drivers' perception of cost of travelling between O-D pairs and waypoints: controlled by the Generalised Cost Equation containing factors of time, distance and monetary tolls
- variability around the perception of cost
- road hierarchy: controlled by the cost of traversing different link types in the network
- dynamic assignment: the re-evaluation of cost (and hence route choice) based on delays occurring during the model period

Dynamic assignment should be enabled where there is any element of reasonable route choice in the study area. Over prescription of routeing is not encouraged when using S-Paramics. For example, turning proportions or lane use figures are not directly programmed into the simulation. Variation of individual link costs is possible but should only be undertaken when the changes can be justified by observation of on-street conditions.

Assignment calibration is an iterative process in that adjustment to the demand matrices or the road hierarchy will affect flows at junctions and hence alter the perceived time delay at junctions. The S-Paramics data analysis tool allows the modeller to readily compare modelled data with surveyed data to rapidly check on the progress of the calibration exercise.

4.7.2 Behaviour Calibration

Calibration of vehicle behaviour within the network to represent observed operation can be split into

- network-wide behaviour
- individual junction/link behaviour

4.7.2.1 Network-Wide Vehicle Behaviour

The development of an S-Paramics model should not normally require alterations to the global parameters which affect vehicle behaviour. Behaviour, in general, alters in response to specific road circumstances and network-wide changes should not be used to calibrate a model unless a sound argument can be made that drivers behave differently in the entire modelled area. The key overall driver behaviour parameters are

- vehicle aggression and awareness distribution
- network headway factor

4.7.2.2 Link and Junction Vehicle Behaviour

For adequate calibration, it is essential that each individual junction in the model operates correctly. With S-Paramics this is achieved by adjusting the node description which controls the paths that vehicles follow through the node and where they stop to give way to each other. Link attributes are also adjusted to control which manoeuvres are permitted from individual lanes and where, as they approach junctions, drivers start to examine opposing traffic flows to find suitable gaps.

Stop lines control the paths followed by vehicles through junctions and where vehicles stop to wait for gaps in opposing traffic. The curvature of the path controls the speed at which the vehicle can traverse the junction; a tight turn requires a lower speed than does a shallow curve. Stop line positions are estimated by S-Paramics as the model is built; complex junctions often require that these automatically generated positions are refined and the road layout adjusted by the modeller.

Each lane entering a junction has a set of permitted turning movements. It may also have a mapping to one or more lanes on the exit arm of the junction. A junction must be calibrated to use the correct turning lanes and movements. S-Paramics makes an initial estimate based on the geometry of the junction, the road types and the available lanes. Complex junctions usually require that these automatically generated lane turning movements and lane maps are refined by the modeller.

Turning movements are prioritised in S-Paramics as ‘major’, ‘medium’ or ‘minor’ in decreasing order of precedence. Major movements have priority, medium movements, usually those which merge with other flows, have the next priority and minor movements, usually those which cross and merge with other flows, have lowest priority. S-Paramics makes an initial estimate of priority based on the link characteristics and geometry. Complex junctions, or those where it is not possible to differentiate road types, usually require that the priorities are refined by the modeller.

Visibility is a key calibration parameter at junctions for medium and minor movements. If the visibility is set to 0 m, then vehicles must stop at the end of the lane at the stop line before they start to determine if there is a gap to move into. If visibility is more than 0 m, then drivers will, at this distance from the junction, start to determine if there is a gap as the vehicle slows to the end of lane stop line. If a gap is present, then the vehicle may not need to come to a halt and may proceed to cross the junction.

The level of visibility is determined by the geographical features of the junction. Figure 4.11 shows two roundabouts. The first has low visibility due to high walls and gradients approaching the roundabout. Vehicles inevitably have to come to a stop as they reach the roundabout before being able to move safely. The second is more open with good visibility. Here, vehicles are less likely to come to a stop before moving out onto the roundabout. The effect on the queues that build up on the roundabout approaches is self–evident.

Fig. 4.11 Comparison of visibility at roundabouts

Links are defined in terms of their physical characteristics such as width, number of lanes and curvature. They are also defined by their statutory characteristics such as speed limits and lane restrictions. As an aid to calibration, there are a small number of further parameters.

There are three gap acceptance parameters: lane merge, lane cross and path cross. These control the gaps that vehicles will accept as they merge or cross into traffic streams. They are normally changed only if there are special circumstances at a junction or drivers are observed to regularly accept small gaps at congested junctions or those with poor visibility. Similarly, default headways may be modified for individual links if evidence is available to support these local modifications.

While S-Paramics does have the ability to code speeds and wait times at the end of links, this is not intended to aid calibration by influencing the queueing behaviour. It is instead intended to model road tables or toll booths. Junction throughput should be calibrated by adjusting vehicle swept paths, stop line positions, opposing vehicle flows and visibility for manoeuvres rather than by including artificial delays or speeds.

Comparison of modelled and observed queue lengths is used to assist in calibration of junction behaviour. Precise comparison can be difficult due to the volatility of the observed data and the subjectivity inherent in deducing if a vehicle is in a queue. The time profile of the demand may be as important to the build up and dissipation of queues as the geometry of the junction.

S-Paramics marks a vehicle as queued based on its speed and the gap between it and the lead vehicle. Different values are used to determine when it leaves its queued state. The values of speed and gap used for marking queued vehicles must agree with those used by on-street measurements to allow valid comparisons between modelled and observed queues. Both the modelled area and the observed area must also extend to cover the likely length of the queue.

4.7.3 Validation

Validation is undertaken when the modeller believes that the model is sufficiently well developed and that further calibration will have limited benefit. Validation requires that the outputs of the model are compared with a set of observed data which have not previously been used in the calibration process. The validation process assures the client that the model is capable of predicting results that have not been explicitly used in the creation of the model and therefore that it adequately reflects reality and is capable of being used to make further predictions.

4.7.3.1 Flows

Link and turn count data can be included in the validation checks. If comparison between the modelled and observed data using the validation set is satisfactory, then it is reasonable to assume that the assignment of traffic in the model adequately reflects reality. If the comparison is not satisfactory, then it should be inferred that while the model has been adjusted to match calibration data, the underlying demand and assignment does not yet reflect the true situation on the road and more work is required to improve the model.

In practice, flow data is rarely withheld from the initial matrix development and calibration process and model validation is undertaken using other measures.

4.7.3.2 Journey Times and Queue Lengths

Journey times may be measured in S-Paramics by defining a path in the model and logging the transit time for every vehicle traversing that path. The same paths are then traversed on the road network or comparable data are obtained via ANPR. The data analysis tool is used to compare the observed journey times with the modelled journey times and undertake the statistical analysis to verify that there are no significant differences.

If differences are found, then the cumulative journey times will assist the modeller to pinpoint the area on the route that caused the validation to fail.

Queue lengths may also be measured along defined paths and compared with observed data. Queue lengths can be quite volatile over short times and observation of length may be truncated in both the modelled and the observed data. Correlation in build up and dispersal of queues, rather than direct comparison of length, can be a valuable and more practical validation tool.

4.8 Extensions

4.8.1 Data Processing

S-Paramics, in common with most microsimulation products, is a Monte Carlo simulation. Each run, with a different random seed, will vary in the same way that each day on a road network is different. Therefore, multiple runs of the base model and the design scheme model are required to reliably assess and compare the effects of implementing a design scheme.

Similarly, microsimulation can provide a vast amount of detailed output on the actions of vehicles in the simulation model. Processing and visualising that data can be a significant task yet it is vital to accurately report on the behaviour of the model.

The S-Paramics data analysis tool (DAT) is a post-simulation data processor that takes the output of a set of microsimulation runs and aggregates and compares the results between a base and a design model.

4.8.1.1 Flows

A partition in DAT is a set of links typically used to compare flows. A partition may by a screen line, a cordon or a set of links associated with a junction. A partition may also be a set of links for which the modeller has survey data.

To analyse the changes in flows in a model, the modeller selects a base model, a design model and a partition to use to select the links for the analysis. DAT then queries the results of the model runs and presents the data as a graph showing a comparison between the mean of the flows aggregated across multiple runs of the model, see Fig. 4.12.

As Paramics is a Monte Carlo simulation, it is to be expected that flows will vary with each model run. Comparison of flows between a base and a design model must therefore include analysis of this random variance to assert if a real change has been made or if the changes are due to random sampling. DAT will plot the mean and the confidence intervals for the runs it has analysed. Examining the overlap between these intervals will help the modeller analyse the significance of the change, see Fig. 4.13.

Flow information can also be manipulated and overlaid on the road network. For example the flows in the base network, aggregated over several runs, may be compared with the flows on a design network, similarly aggregated and the difference plotted to show how the change in the road network has affected the assignment of vehicles to the rest of the network, as shown in Fig. 4.14.

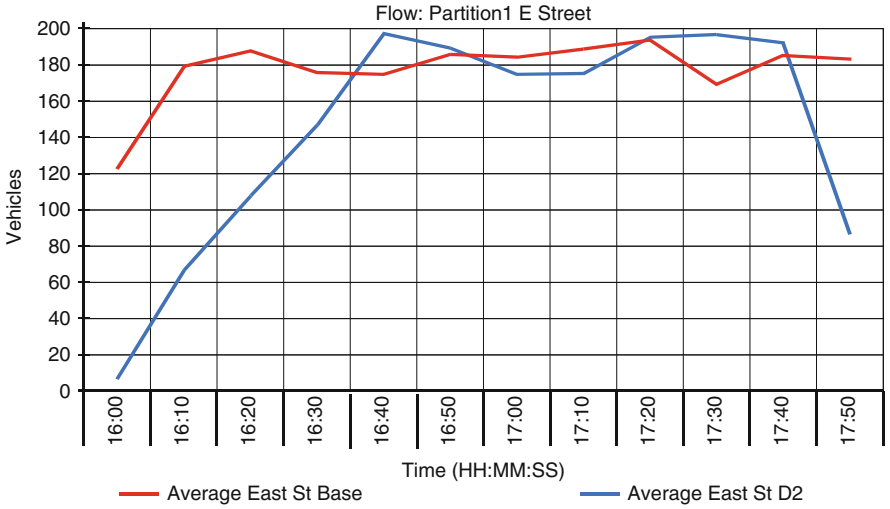


Fig. 4.12 Flow comparison

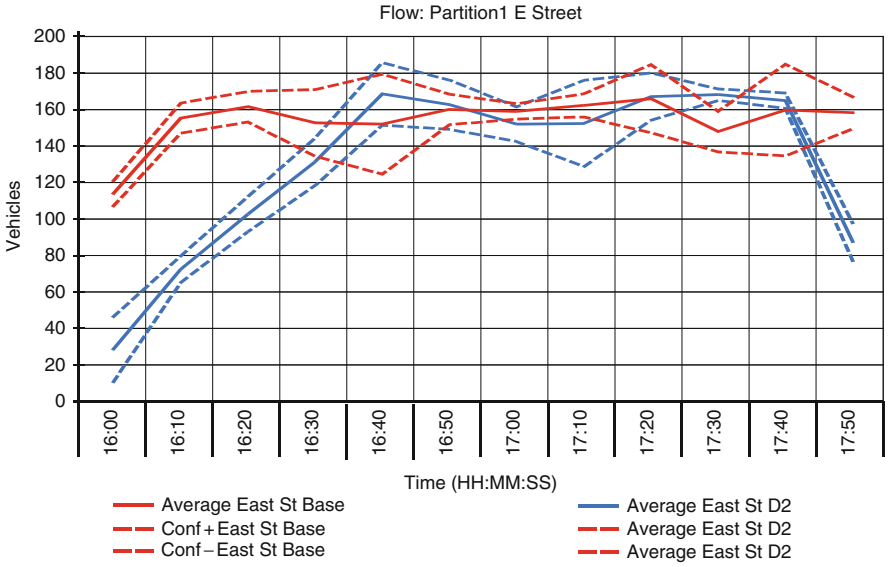


Fig. 4.13 Flow comparison with confidence intervals

4.8.1.2 Queues

S-Paramics will mark vehicles that slow down to queue behind another and will then write in the output the length (in metres and in vehicles) of the observed queues at the collection interval as the simulation runs. Queues can be measured along a

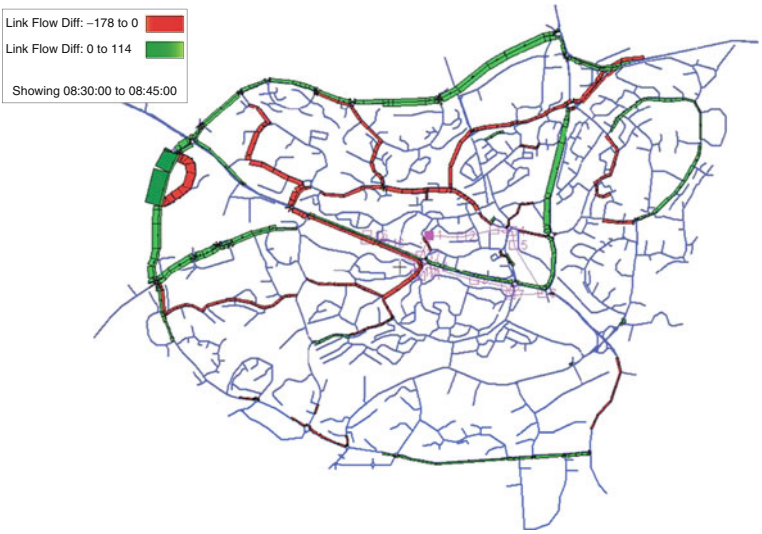


Fig. 4.14 Flow differences

pre-defined path or can be displayed on each link. With each analysis the modeller is able to select minimum, maximum and average queue length over the collection interval in terms of metres, vehicles or PCUs. The average can also be computed for the whole collection interval or just for the time where a queue is present. Figure 4.15 shows a queue path towards a junction, crossing a second minor junction. The first and last links are specified; the intermediate links are derived from the shortest path. The graph shows the maximum, mean and minimum queue length by time of day.

4.8.1.3 Journey Times

S-Paramics can measure journey times in two ways: for the whole trip from origin to destination and along a determined path in the model. As before, the addition

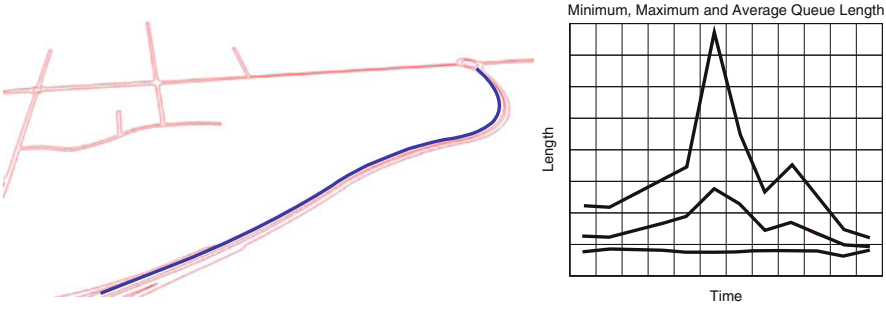


Fig. 4.15 Queue length summary

of confidence intervals allows the modeller to compare journey times between a base and a design model and start to analyse the effect of the change. Similarly the minimum and maximum and standard deviation of the journey times can be aggregated across multiple runs to analyse journey time variation.

4.8.1.4 Events

S-Paramics also logs events such as lane change, overtaking manoeuvres, etc. DAT may be used to plot these on a skeleton of the network to analyse where excessive weaving takes place or where overtaking is occurring. Figure 4.16 shows a map of the model with lane change events plotted in 30 m intervals along the road. The location of weaving sections is made clear.

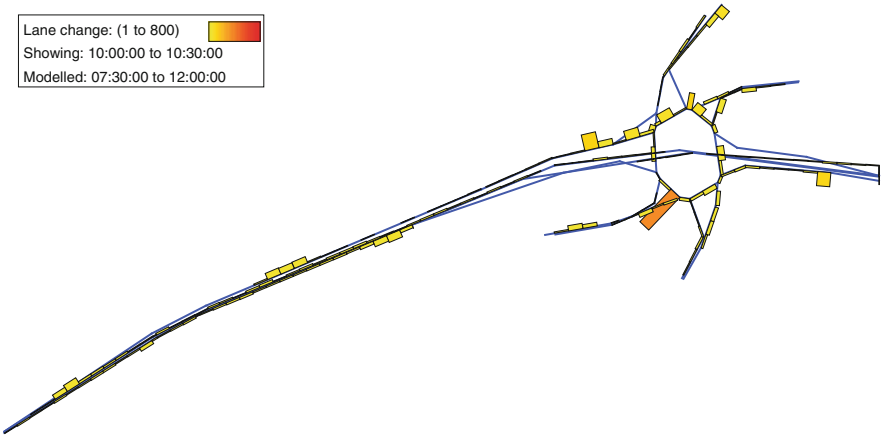


Fig. 4.16 Event plot showing lane change events

4.8.2 Batch Farm

DAT allows the modeller to readily process the data from multiple model runs. The batch farm is a mechanism for generating that data from multiple runs in minimal time and with maximum use of available resources.

The batch farm uses grid computing techniques to enable multiple computers to be assigned to the task of running S-Paramics simulations. S-Paramics has a 'batch' mode in which all the vehicle trips, individual behaviours, movements and routeing decisions are undertaken as usual and the statistics output created. None of the presentation or visualisation is present, hence the model runs many times faster than it does when running in visualisation mode.

The batch farm accepts a request from the modeller for *N* runs of a particular model. Each participating computer polls the batch queue and runs waiting simulations. The modeller is informed when all the runs are complete and the data consolidated in one location.

The batch farm can be extended to include many PCs; some modellers install a set of dedicated processors to run the farm, while others use idle PCs overnight. S-Paramics licences can be extended to include multiple batch processes to allow extended batch farm use.

4.8.3 PEARS

PEARS (Program for the Economic Assessment of Road Schemes) is an economic assessment package that has been designed for use with the output from traffic microsimulation models. The methodologies and costs are derived from the UK TAG Unit 3.5.6 – Values of Time and Operating Costs (www.webtag.org.uk).

PEARS carries out trip-based assessments of changes in travel time costs and vehicle operating costs. The costs of a trip-based assessment are derived by aggregating the costs of each individually modelled vehicle on the network. Using microsimulation to provide the trip-based data for economic assessment adds robustness to the assessment as it can include

- detailed network definition
- flexibility in modelling the operating characteristics of different vehicle types
- accurate representation of the variation in traffic flows using demand time profiles
- detailed modelling of individual vehicles particularly in platooning and overtaking
- more accurate assessment based on emissions and fuel consumption based on the details of a vehicle's speed and acceleration during the trip

The output from PEARS is in the form of a set of standardised tables describing the cost profile of the scheme, the value of the benefits of the scheme aggregated over time and discounted to a base year and the benefit-to-cost ratios used to help prioritise schemes as part of a budget and planning task.

4.8.4 Signal Control and ITS

4.8.4.1 Advanced Control Interface (ACI)

The ACI is a means of gathering data from collection devices within the simulation and using the data to control the actions of simulated vehicles. The ACI was designed to allow the inclusion of adaptive signal control and ITS systems within an S-Paramics simulation through 'hardware in the loop' simulation and through

'software in the loop' simulation where the software to be included is the kernel of the hardware control system.

The ACI is based on SNMP (simple network management protocol), a communication standard in common use by UTC system engineers. SNMP allows multiple connections to the simulation and cross platform, cross network operation as well as using communications protocols already in use by UTC suppliers.

The data exchanged by the ACI closely mimic those available to current UTC systems. These are

- inductive loop data: speed flow and occupancy from traditional detector loops
- journey time data: flows and speeds along known paths – in effect a proxy for ANPR camera technology
- emissions data: the results of a roadside emission detection device
- congestion detector: measurement of queue lengths
- car park occupancy: the number of spaces and the capacity of each car park

The actions that can be implemented by the ACI are also designed to closely follow those available to UTC system engineers.

Traffic signals. Stage times may be set for a single use in the next cycle or for continued use over multiple cycles. Hurry calls for stages may also be made to terminate the existing stage and move to another; stages may be run in or out of order. For signal controllers which are not stage based, movements may be designated as belonging to signal groups and these groups controlled through the ACI.

Variable message signs (VMS). VMS are placed at the roadside or over the road on gantries. Vehicles are given information as they pass the sign. Broadcast devices are also available through the ACI. These pass messages to all drivers in the simulation or to a subset in an area of the model. These devices are intended to act as a proxy for in-car receivers such as radios or SatNav devices.

The messages passed through VMS and broadcast devices consist of three parts:

- the message as seen on the gantry, or as received in car – may be an image file or a free text
- a formal interpretation of that message used to affect the behaviour of drivers in the simulation
- a response profile specifying which drivers will act on the information in the message

The actions which can be affected by these messages are the following:

- speed, either as a target with the same variability as a normal road speed limit or as a mandatory maximum speed regardless of driver aggression levels
- headway modification
- lane restrictions
- changes in aggression and awareness

- delay warnings to inform drivers of congestion and include this extra delay in their route calculations
- diversion instructions to divert drivers via a named waypoint
- car park advice to change a driver's destination to a specified car park

The response profiles may select drivers based on

- vehicle type, e.g. HGVs – selection may be inclusive or exclusive
- driver awareness and aggression levels – selection may be based on a greater than or less than comparison
- random factor or fixed percentage of vehicles
- vehicle destination zone or destination car park

A single message may have multiple formal interpretations and multiple response profiles. For example a message posted as '20 Minute Delay at New Town Junction' may be interpreted by the optimistic drivers as a 10-min delay, by most as a 20-min delay and by the more pessimistic as a 30-min delay.

Several applications have been built using the ACI as well as a number of ad hoc signal controllers, signal optimisers and ITS systems.

4.8.4.2 ACI Example: Automated Traffic Management

Variable message signs (VMS) control the flow of vehicles on motorways and automated traffic management (ATM) can be used to provide the method of control indicated by these signs. The ATM controller is a software that links to an S-Paramics simulation to control VMS in the simulation. The ATM controller implements the Motorway Incident Detection and Signalling (MIDAS) system as documented in HA report NMCS2 MIDAS Signal and Sign Setting Strategy: MCH 1744 (Issue F, July 2004). It also implements ramp metering as documented in HA report RAMP Metering System Requirements Specification: MCH 1965 (Issue D July 2008).

Automated motorway management came to prominence in the UK in reaction to a proposal in 2001 to widen the M25 London orbital motorway to 14 lanes in its busiest sections. This option was considered to be untenable and as a result, more active control of motorways was planned. This control required intelligent setting of speed limits, access control through ramp metering and protection of congested areas to reduce the propagation of congestion-based shock waves.

MIDAS uses induction loops at 500 m intervals to detect slow-moving traffic arising from an incident or from congestion. MIDAS then sets speed limit signs to 60, 50 and 40 on gantries upstream of the incident to progressively slow traffic approaching the back of the queue. This protects the back of the queue from secondary incidents as well as reducing the backward propagation of the congestion shock wave.

The same VMS gantries may be used to indicate lane closures, lane-specific speed limits or vehicle-type-specific restrictions. These are used to test strategies for incident management.

The ATM controller may also be used to implement hard shoulder running with its associated speed restrictions and examine the relative benefits of adding the extra lane against reducing the speed limit. This reduction is required as the hard shoulder is often constructed to a lower standard than the main carriageway.

Ramp metering reduces the demand on motorways by controlling the flow of vehicles entering an already congested main carriageway. The ramp metering algorithms in ATM include the Alinea and demand capacity algorithms. Queue protection and queue override are also available and further access management techniques are planned. ATM is able to examine the result of adding ramp meter controls in conjunction with active control of the main carriageway (Fig. 4.17).

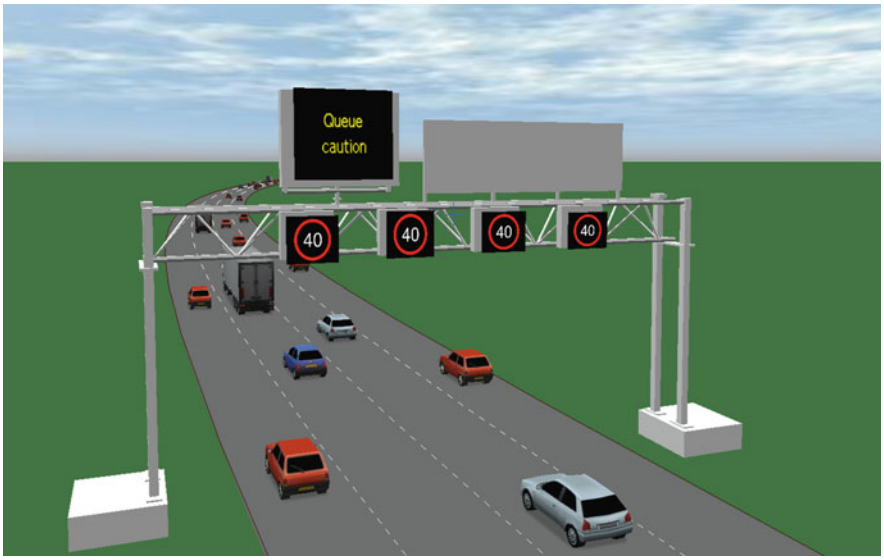


Fig. 4.17 VMS gantry controlled by ATM

4.8.4.3 ACI Example: UTC Signal Control

S-Paramics has links to several urban traffic control (UTC) systems. These enable the simulation to include adaptive signal control from systems such as SCOOT and SCATS.

Data from vehicle detectors in the simulation are relayed to the UTC system which aggregates the data from many detectors. The UTC then uses its own internal model of the road network and determines when signals are to change to optimise flows through the network. The instructions to the signals are relayed back to the simulation model where the stage changes are made.

The system architecture for both SCOOT and SCATS uses a software only implementation of a 'hardware in the loop' simulation. The same core UTC software is used to provide the control algorithms and network model but the interface layer of the software, which would normally communicate with roadside 'outstations', is replaced with an interface layer that communicates with the simulation via the ACI. Collaboration with the UTC vendors has been key to this implementation.

4.8.4.4 ACI Example: Signal Controller

MOVA from TRL (Transport Research Laboratory) in the UK is a single-junction or small area adaptive signal control system. It is often embedded in an 'outstation', an on-street signal controller. PCMOVA is the MOVA kernel running on a PC and linked to microsimulation. It is used to include the adaptive signal control algorithms within the simulation and hence test the effectiveness of MOVA or the effect of changes to the MOVA data set. CCOL, a signal control system used in The Netherlands, operates in a similar way.

The Sentinel signal controller from Telent in the UK is another example of a signal control outstation which has been linked, using the ACI, to S-Paramics. Once again this was done as a software implementation of a hardware in the loop simulation. The Sentinel controller may include MOVA as well as the usual static signal stage timing logic, vehicle actuation of signals and pedestrian stage calls.

4.9 Case Studies

4.9.1 Large Models

4.9.1.1 Plymouth

SIAS was commissioned by Plymouth City Council to develop an S-Paramics microsimulation model of Plymouth and the surrounding area. The model includes 230 zones, 454 km of road network and spans some 35 km × 20 km. It also includes the location of 1,300 bus stops within its description of every scheduled city bus service. Beyond the city boundaries, it extends to Liskeard, Tavistock and Ivybridge.

The Plymouth model contains many of the most complex features found in other S-Paramics models and some additional facets unique to the city (Fig. 4.18). For instance, the Torpoint ferry service has been simulated to include both the full operation of the ferry service and the queueing and vehicle marshalling areas. The checkpoints at the naval dockyards are also included, where the effects of occasional periods of heightened security can be assessed for their impact on the rest of the road network.

With its S-Paramics model, Plymouth City Council has taken steps towards holistic transport impact assessment and traffic management. The model has been validated to a high standard to ensure that land developments within the entire Plymouth area can be assessed within the same unified framework. This enables

Fig. 4.18 Plymouth model area

transport-planning priorities, phasing of road works, local and remote traffic impacts and the most appropriate application of ameliorative measures to be assessed and applied.

4.9.1.2 Chelmsford

This is one of the largest microsimulation models in the UK, extending from the A120 in the north to the Dartford crossing in the south (Fig. 4.19). The original purpose of the model was to test options for the north-eastern bypass, but it has now been used to test the effect of various traffic management design options within Chelmsford, including the installation of pedestrian signalised crossings. Over 90,000 vehicles are represented in each of the weekday peak periods, with up to 12,000 vehicles simultaneously modelled. The model includes a directional flyover at the Army & Navy roundabout, where the flow direction reverses at 14:30. The model includes the highly successful park and ride site at Sandon, opened in April 2006 and now extended to 900 parking spaces.

4.9.1.3 Alkmaar

In the Alkmaar, North Holland region, the national, provincial and regional road organisations jointly commissioned a wide area simulation model to assist in central coordination of the large number of projects and events occurring in the region (Fig. 4.20). By visualising cumulative effects, the authorities could gain insight into the interaction of different combinations of projects and could consequently make

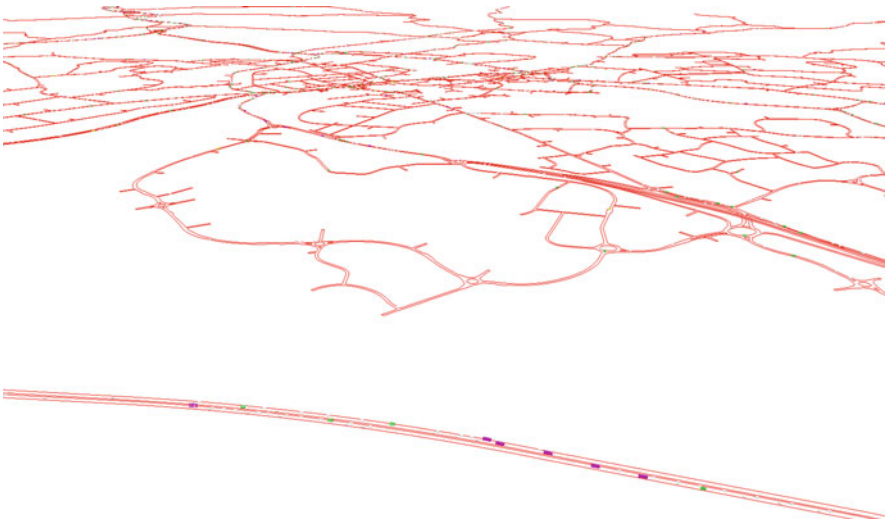


Fig. 4.19 Chelmsford model area

Fig. 4.20 Alkmaar model

informed decisions. The decision was made to use a microsimulation model because of its capabilities for simulating temporary road conditions in a realistic manner, such as lane closures, changes to junctions and dynamic route choice through ITS. The ability of microsimulation to record vehicle-specific travel times was required in establishing police and fire service travel times during construction and maintenance periods.

The simulation model provided detailed insight into the effects of road works and events. The forecast effects and the duration of road works are stored in a database. This data includes the re-routeing observed in the model, the initial mitigation measures planned for an event and any extra measures found to be required during testing. The database is used to assist in planning decisions and is continually updated and expanded.

4.9.2 UTC and ITS

4.9.2.1 Hampton Court Flower Show

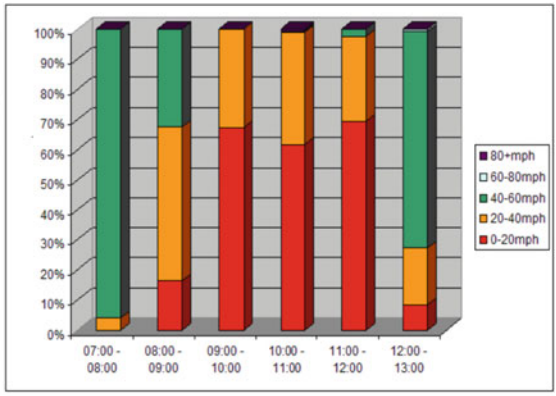
Hampton Court Palace Flower Show is an annual event organised by the Royal Horticultural Society (RHS). Sited at Hampton Court Palace in the south west of Greater London, it is the world's largest annual flower show and takes place in early July over a period of 6 days. On average, 179,000 visitors attend the event each year, approximately 30,000 per day. Although this is an average of only 3,500 vehicles per day, the difficulty is caused by the peaked nature of the flow.

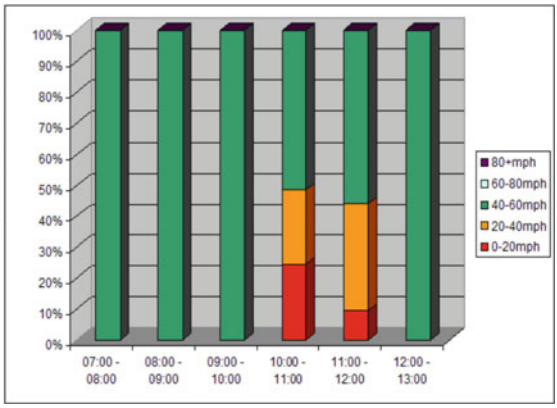
The main problem is that a traffic queue extends back from the nearby town of Esher to the junction with the A3 trunk road. This becomes critical when the queue extends onto the A3 main carriageway resulting in erratic driver behaviour and dangerous driving conditions for those travelling on the A3. Therefore, in 2008, a set of SCOOT UTC strategies were developed by Surrey County Council to reduce the queuing. Each of these were tested using the S-Paramics model of Esher and the link to the UTC system operating in Esher.

Strategies were devised using a Delphi technique to list and initially screen the options. Four were tested with the simulation model, the development of each involving an iterative process of defining, simulating and evaluating the effect of individual parameters. Data analysis using output from S-Paramics and from SCOOT was used to evaluate the effectiveness of each change. The overall effectiveness of each strategy was evaluated using a composite measure including air quality and safety measures. The latter was determined by observing how many vehicles were queued at the junction with the A3.

The conclusions reached by Surrey County Council after the exercise were that one of the greatest advantages of testing strategies in a microsimulation environment was the ability to observe and assess its effects on the entire network at any one time through its visualisation. This led to a quick dismissal of strategies that did not meet the objective or demonstrated major disadvantages to the rest of the network. It also enabled several strategies to be developed and tested quickly within a controlled environment, which would not have been possible if the same strategies had been trialled on-street during the same period of time, with or without event conditions. During the event the chosen strategy, the SCOOT technique of gating traffic through Esher, met the safety and throughput objectives and showed strong correlation with the prediction from the model. Also, despite close scrutiny by CCTV during the event, no changes in the UTC data sets were required and no complaints from the

Queue at A3 northbound exit in 2007





Vehicle speeds at A3 northbound exit slip at Esher Common on the second day of the show in 2007 and 2008. Data from HA Traffic Information Database.

Fig. 4.21 A3 at Esher

public were received. Figure 4.21 shows the situation in 2007 with the queue extending on to the A3 trunk road. The graphs compare vehicle speeds at the top of the slip road in 2007 and 2008 and show the reduction in queueing in this critical region due to the changes in the UTC system.

4.9.2.2 M25

An S-Paramics microsimulation model of the controlled section of the M25 between junctions 10 (A3) and 16 (M40) was developed to allow testing of motorway control strategies (Fig. 4.22). This was done through modifications to the Motorway Incident Detection and Automatic Signalling (MIDAS) system that currently manages traffic on the M25.

The operation of MIDAS was incorporated into the model by linking it to the ATM controller. This links to overhead gantries and traffic detector loops in the simulation to replicate the MIDAS system through the generation of speed controls and automatic signalling in response to congestion and/or incidents. Changes to

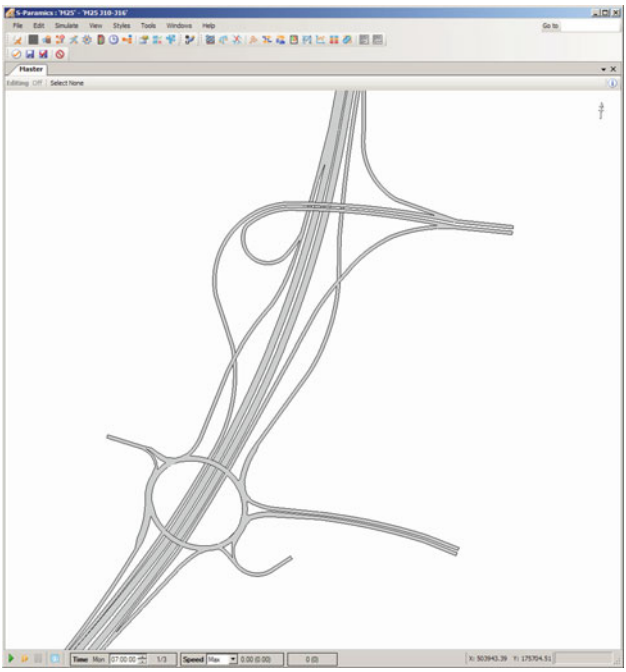


Fig. 4.22 Sample M25 junction

the MIDAS specification were tested and the results showed an improvement in journey times and more importantly, an improvement in journey time reliability, a key performance indicator for the M25.

4.9.2.3 Car Park Guidance

In 2008, the town of Nieuwegein, south of Utrecht in The Netherlands, was experiencing rapid growth and plans were drafted to develop the city centre. One aspect of this development was in increase in city centre parking from 2,759 places to 4,669. The city planners commissioned an S-Paramics model to investigate the effect of the increased flows into the town centre and also creation of a city centre pedestrian zone. This investigation included the design of an ITS parking advisor to direct vehicles to car parks on the basis of capacity and priority, thus reducing the volume of traffic within the town by removing the journeys between car parks while hunting for a space.

The simulation model was built to include multiple entries and exits from each car park. Each car park served multiple origin and destination zones and each zone could be served by multiple car parks. Complete trips were modelled in the simulation run. Vehicles arriving at a car park were constrained to subsequently leave

Fig. 4.23 Car park occupancy

from the same car park on their return journey. The base model validation included comparisons between observed and modelled car park occupancies.

Figure 4.23 shows the effect of the ITS system on the car park occupancy near the end of the simulation run with and without the ITS guidance system. Vehicles are more evenly distributed and spaces are available in all car parks. There is an overall benefit, for drivers both observing and not observing the advice from the ITS car park advisor, in the reduction in the need to hunt for a parking space.

4.9.3 Road Design Studies

4.9.3.1 Overtaking Study

On a single carriageway, a number of vehicles may be slowed by a single slow lead vehicle, typically an HGV or an agricultural vehicle. Consequently, vehicles will form a platoon behind the lead vehicle for as long as they are unable to pass it. If an improvement scheme includes carriageway widening to facilitate dedicated or opportunistic overtaking, then the benefit of the scheme will largely be felt by the vehicles that can now pass the lead vehicle, an effect known as platoon dispersion. This benefit is not felt just over the length of the scheme but for some distance downstream.

S-Paramics includes an overtaking model in which a vehicle will first assess its desire to overtake based on its target speed with respect to the speed of the vehicle ahead. It will then assess its ability to overtake based on the gap available ahead of the vehicle to be overtaken and also the visibility of the road ahead and absence of an oncoming vehicle in that space. The overtaking manoeuvre is initiated when both desire and gaps are present.

The Scottish Executive, working on behalf of all UK government transportation departments, commissioned a study to assist with the preparation of a new technical advice note for provision of overtaking lanes. The study specifically examined platoon formation, the effectiveness of different lengths of overtaking lane and the extent of the downstream benefits.

A simulation model was developed to replicate a 15-km section of single carriageway. The model was coded to include three sections: an approach length, a central study section followed by a run-out section. The run-out, or downstream section, of 10 km allowed the model to measure the downstream benefits. Calibration included adjusting the overtaking parameters such that the change in vehicle order in the model matched that measured from a set of number plate matching surveys. The mode was used to investigate the effect of adding WS2+1 sections (wide single carriageway, two lanes plus one overtaking lane) and comparing this with existing opposed overtaking (Fig. 4.24).

Fig. 4.24 Overtaking model on WS2+1 section

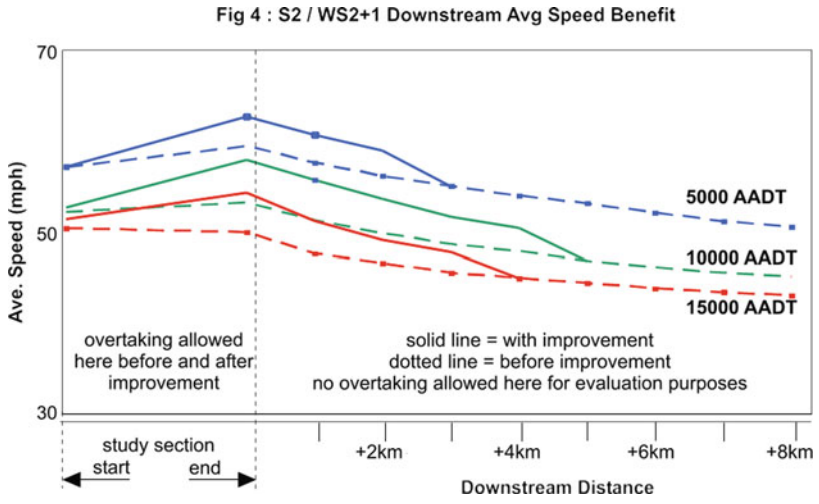


Fig. 4.25 Overtaking downstream benefit

The graphs in Fig. 4.25 show the average speed of vehicles under three different flow rates, with and without the overtaking section. The results indicate that

- at 10,000 AADT, the benefit in increased vehicle speed is accrued for around 5 km downstream
- at 5,000 AADT, the benefit is of shorter duration as the overtaking demand is satisfied earlier
- at 15,000 AADT, benefit is again of short duration as overtaking vehicles soon catch up on next platoon

S-Paramics is able to capture the economic benefits derived through the provision of overtaking lanes through detailed simulation of individual vehicle behaviour, specifically their variation in aggressiveness which leads to speed differentials and overtaking. Platoon formation is implicit under such circumstances and essential to the analysis of the economic benefits of the provision of overtaking sections.

Chapter 5
Traffic Simulation with Aimsun

Jordi Casas, Jaime L. Ferrer, David Garcia, Josep Perarnau, and Alex Torday

5.1 Introduction

5.1.1 Background and Overview

Originally the focus of a long-term research programme at the University of Catalonia (UPC), the Aimsun transport modelling software is now in its sixth major commercial version. Having outgrown the stated aim of the original AIMSUN acronym 'advanced interactive microscopic simulator for urban and non-urban networks' (Ferrer and Barceló, 1993; Barceló et al., 1994, 1998a), the software now includes macroscopic, mesoscopic and microscopic models and is simply known as 'Aimsun' (Aimsun, 2008).

Expanding in response to practitioners' requirements, Aimsun 6 has come to encompass a collection of dynamic modelling tools. Specifically, these include mesoscopic and microscopic simulators and dynamic traffic assignment models based on either user equilibrium or stochastic route choice. From a practitioner's standpoint, macroscopic modelling plays an increasingly important role in the area of demand data preparation. However, in line with the scope of this book, this chapter focuses on Aimsun's dynamic modelling capabilities.

The primary areas of application for Aimsun are offline traffic engineering and, more recently, online (real-time) traffic management decision support. In either case, the use of Aimsun or Aimsun Online aims to provide solutions to short and medium-term planning and operational problems for which the dynamic and disaggregate models described in this chapter are extremely well suited. Strategic planning is an

J. Casas (✉)
TSS – Transport Simulation Systems, S.L, Passeig de Gràcia 12, 08007 Barcelona, Spain and Universitat de Vic, Carrer Sagrada Familia 7, 08500, Vic, Spain
e-mail: casas@aimsun.com

J.L. Ferrer, D. Garcia, J. Perarnau, and A. Torday
TSS – Transport Simulation Systems, S.L, Passeig de Gràcia 12, 08007 Barcelona, Spain
e-mails: jlferrer@aimsun.com; david@aimsun.com; josep@aimsun.com; torday@aimsun.com

J. Barceló (ed.), *Fundamentals of Traffic Simulation*,
International Series in Operations Research & Management Science 145,
DOI 10.1007/978-1-4419-6142-6_5,

adjacent realm for which more aggregate and/or static models continue to be very suitable. There are important interfaces between those two realms at the level of methodology (effect on demand of lasting changes to the effective capacity) and technology (importing from and exporting data to strategic planning software) and we will comment on those issues further in the coming sections.

5.1.2 Development Principles

The remainder of this chapter will provide details on the models inside Aimsun as they currently stand. Inevitably, this description can only be a snapshot of what is currently available with very limited references to ongoing developments. The fast-paced evolution of this software category in general, and of Aimsun in particular, almost guarantees that some aspects of our description will be obsolete or, at best, incomplete soon after publication. It is therefore worth providing the reader with a brief outline of the overarching principles that will continue to inform the development direction of our transport modelling software platform beyond its sixth major release.

- *Integration*: The steadily improving volume and quality of traffic data, availability of computing resources and, perhaps most importantly, practitioner's expertise have given rise to sophisticated and rich methodological frameworks for dynamic modelling. A key exponent of this tendency is the integrated corridor management initiative described in Alexiadis (2007) and put into practice throughout North America and elsewhere (Stogios et al., 2008; Torday et al., 2009). Models are steadily growing in size, complexity and detail; and feedback loops between demand and supply changes are explicitly acknowledged in modelling studies. This development naturally calls for elimination of duplicate information that, in that context, represents wasted effort and risk of error.

  Concretely, in the case of Aimsun, the information shared by all models is network topology, OD demand and time-dependent shortest (or cheapest) paths and their respective travel times (or costs). Demand and paths/path costs are not merely a shared input; it is rather the case that the application of one model can produce those outputs in a format that is directly exploitable by another Aimsun model. The advantages of this approach are further elaborated in the case studies discussed in Section 5.7. The underlying principle is to integrate everything that can and should be shared between all the models to enable sophisticated workflows that involve sequential, iterative or even concurrent application of two or more models. The other implication of the principle is *completeness*; put simply, this translates into 'integrate everything that is required to meet a modelling study's objectives comprehensively'.
- *Modularity*: This refers to the breaking down of processes or tasks into elementary units, allowing their consistent and easy re-use within larger processes. Examples of how this principle has been applied in Aimsun include sharing

the gap acceptance or traffic management modules between the mesoscopic and microscopic simulators. Perhaps more crucially, an insistence on modularity allows Aimsun users to decouple the link dynamics models (i.e. the mesoscopic and microscopic simulators) from the process by which traffic is assigned dynamically to various routes. This gives rise to two new combinations of models with potentially very interesting and useful applications. Firstly, the combination of mesoscopic simulation with dynamic traffic assignment by means of locally applied stochastic route choice can be a great tool for faster than-real-time simulation of non-recurring incidents – especially so when it is also possible to specify that drivers away from the incident's influence area will continue to follow established routes (the results of a previous dynamic equilibrium assignment). Secondly, the combination of dynamic user equilibrium and microscopic simulation can shed a lot of light onto town-centre modelling in which modal shift policies (pedestrianisation, dedicated bus lanes, public transport pre-emption schemes) bring about lasting changes to network utilisation and in which pedestrians, public transport and private vehicles interact at stops and crossings creating mutual dependencies that affect capacity in complex ways. The underlying principle is to break models down into basic ingredients such that practitioners can re-combine these ingredients as appropriate into new methodological 'recipes' (best practice).

- *Scalability*: Computing hardware continues to improve every year. ITS and other developments related to communication follow at an equally fast pace, leading to a parallel improvement in the availability and quality of traffic data. The two factors combined push the envelope: dynamic modelling is now considered desirable and useful in areas such as the entire area of lower Manhattan, New York; Montreal in Quebec or all of Singapore. Fortunately, it is also feasible using mesoscopic simulation or multi-threaded implementations (Barceló et al., 1998b) of micro-simulation or a combination of both (Barceló et al., 2006). The challenge in the years to come will be to keep apace with changing intensive computation paradigms whilst responding to ever more stringent performance requirements imposed by users. Efficient software design is, in our opinion, a prerogative in that regard.
- *Interoperability and extensibility*: The proliferation of ITS systems on the one hand and GIS/mapping and 3D technologies on the other means that creating an accurate model of a road network is becoming much more difficult and quite a lot easier at the same time. Variable speed control, dynamic lane assignment, congestion pricing and adaptive signal control are examples of novelties that need to be captured accurately in models. Conversely, high-quality maps and GIS data simplify model building compared to a few years ago; as for 3D building information, it is making the prospect of large-scale virtual reality viable. Whatever the case may be, all these developments call for *interoperability*, that is the ability to exchange data with other applications in a variety of formats, and *extensibility*, that is the ability for users to programme custom extensions relatively easily. Again, we note that the ability to respond to these challenges rests, to a large

degree, with efficient software design although *standards* play an important role here as well.

The remainder of this chapter is organised as follows:

- Model-building principles in Aimsun
- Fundamental core models: car following and lane changing
- Dynamic traffic assignment
- Calibration and validation of Aimsun models
- Extended modelling capabilities: working with external applications
- Selected overview of advanced case studies and applications
- Modelling details of advanced case studies

5.2 Model-Building Principles in Aimsun

Building a transport simulation model with Aimsun is an iterative process that comprises three steps:

- Model building, that is, the process of gathering and processing the inputs to create the model;
- Model verification, calibration and validation, that is the process of confirming that implementation of the model logic is correct; setting appropriate values for the parameters and comparing the outputs of the model to corresponding real-world measurements in order to test its validity;
- Output analysis, that is the exploitation of model outputs in line with the overall objectives of the modelling study.

The following three sub-sections provide further detail on these steps.

5.2.1 Model Building

Building an Aimsun model requires two types of information:

- *supply data*, that is everything related to the infrastructure and services that allow goods and people to travel coded as a graph of sections and turns with associated attributes;
- *demand data*, that is the mobility needs, coded as a set of OD matrices, one for each vehicle type and time interval.

Calibration and validation bring about additional data requirements at both the supply and the demand level. This can include information about the types of vehicles (e.g. acceleration characteristics), drivers (e.g. level of compliance with speed

limit) and actual levels of traffic through the network for known levels of demand. Accordingly, Aimsun supports the concept of real data sets and allows the user to manage them.

5.2.1.1 Supply Data

Supply data includes all the information related to the transportation network and services, such as

- geometric and functional specification of the road network;
- traffic control;
- public transport services;
- other (for instance, fleet vehicles).

5.2.1.2 Geometric and Functional Specification of the Road Network

Geometric information that is needed to build an Aimsun model are the following:

- road shape;
- number of lanes;
- reserved lanes;
- turnings allowed at the end of each section – from which lane(s) to which lane(s), together with stop points and priorities between conflicting movements;
- pedestrian crossings.

Functional attributes (some of which present in all geographic information systems) depend on the level (micro, meso or macro) of simulation (Table 5.1).

Table 5.1 Physical parameters required for each model type in Aimsun

	Macro		Meso		Micro	
Parameter	Required	Optional	Required	Optional	Required	Optional
Maximum speed	X		X		X	
Capacity	X			X[a]		X[a]
User-defined costs		X		X		X
Volume delay function	X					
Visibility distance at yield intersections					X	
Slope	X		X		X	

[a]Can be used as an additional parameter for route choice

The process of coding the network uses as its basis a file in one of the following formats imported into Aimsun:

- aerial images, such as PNG, JPG, BMP, GIF, SVG, SID, ECW, JP2 and TIFF;
- 3D models, such as 3dsmax 3DS and Wavefront OBJ;
- CAD, such as AutoCAD DWG or DXF, and Microstation DGN;
- GIS, such as ESRI SHP, MapInfo TAB or MIF, OpenGIS GML, GPX and Google KML;
- digital maps, such as Navteq maps files;
- input files for other transport modelling or signal optimization software applications, such as Emme, SATURN, CONTRAM, VISUM, VISSIM, PARAMICS, TRANSYT and SYNCHRO.

The amount of information that can be automatically converted, and thus the amount of manual refinement that is needed, depends on the type of model that must be built (macro, meso or micro) and on the format of the input file. The file formats above are listed roughly in the order of ascending utility for modelling purposes (i.e. the most useful format is listed last).

For example aerial images, CAD files and 3D models do not carry any kind of topology or functional information, so they can only be used as background to guide manual network building; some image formats do not even provide geographical location or scale.

GIS files, maps and other traffic simulation software files are the best formats because they include both geometry and additional attributes associated with each entity, so the import process automatically creates a complete Aimsun network; but even in this case a subsequent manual refinement is needed, for example to locally adjust node details and to input a parameter required for Aimsun simulations for which there is no equivalent in the third-party software.

5.2.1.3 Traffic Control

For microscopic and mesoscopic simulation, traffic control plans during the simulation period are needed for all signalized intersections and for any ramp meterings included in the model.

Traffic signals can be fixed, i.e. defined in advanced and remain unchanged during the simulation period, or actuated, whereby the control plan is dynamically modified depending on measured traffic conditions.

For each fixed control plan, Aimsun requires the following information:

- start time and duration of the control plan;
- cycle length;
- amber/yellow duration;
- turnings associated with each signal group (including pedestrian movements);

- timings of each signal group;
- offset relative to other control plans.

Actuated control plans can be modelled using virtual detectors which are the simulation equivalents of real-world loop detectors or other similar devices. As for the control logic, this can be specified in Aimsun, if the real controller is compliant to the NEMA standard. With SCATS, UTOPIA, VS-PLUS, SICE and SCOOT, a data interchange interface can be used; otherwise the control logic can be emulated by programming a custom API (application programming interface) extension which exchanges data between Aimsun and the software implementation of the corresponding controller logic.

5.2.1.4 Public Transport

In order to include public transport into the model, the following information must be provided:

- route of each line;
- stop locations;
- departure frequency or timetable;
- stop-time mean and deviation. This can be global or, as an option, a function of the line, stop and time of the day.

5.2.1.5 Demand Data

Traffic demand is input into Aimsun in the form of either (time-dependent) OD matrices or traffic states (micro only).

If OD matrices are used (recommended for micro and required for meso), it is necessary to have the zoning of the modelled area to correctly place the centroids and their connections. The placement of centroid connections has to be carefully studied so that the entry and exit rate into and from the model is as realistic as possible. To minimize this distortion, internal centroids should never be connected to main streets but to local roads and preferably at nodes rather than to specific sections.

In order to reproduce traffic patterns and fluctuations faithfully, it is advisable to input separate OD matrices for different vehicle types using small time slices (possibly 15 min).

Similarly with all models of this category, recent and reliable matrices are a prerequisite: to that end, a matrix adjustment using traffic counts is often helpful for improving the quality of an old matrix if a more recent one is not available – provided that no big changes have occurred in land use.

OD matrices can be pasted into Aimsun via a simple copy operation in Microsoft Excel or be read directly from an ASCII file or any database via an ODBC connection.

If traffic states are used (possible only for micro-simulation), users need to specify the input flows for all entrance sections and the turning percentages at each node where more than one turning is possible.

Simulating with traffic states is generally not recommended. Firstly, for some network configurations, it is possible that the sampling process at nodes will lead to one or more particular vehicles becoming 'trapped in' the network (that is they move around continuously and never exit); so it is acceptable only for small networks whose connectivity does not lend itself to loops. Secondly, for all types of networks, traffic states cannot be used to simulate a future scenario in which a change to the supply or the demand may create different route utilization.

5.2.2 Model Verification, Calibration and Validation

Before starting to modify model parameters in order to calibrate the model, the user must be sure that there are no specification errors that affect the model logic and therefore simulation results.

Verification consists in assuring that the model has been correctly edited in Aimsun, checking network geometry, control plans, management strategies and traffic demand, and verifying that the model description corresponds to the objectives of the study.

Aimsun provides a tool that can automatically detect errors in supply definition, such as a section where not all the lanes at the beginning or at the end are connected or an OD pair with trips but no feasible path.

Verification of traffic demand is done through a manual comparison with traffic counts wherever possible; for example the total trips generated and attracted by a zone must be compared with the counts of the sections to which the corresponding centroid is connected.

An important check is to verify that the model is suitable for the objectives of the study; the model must include all the area that might be influenced by future changes being modelled; the boundaries must be free of congestion; if rerouting strategies are simulated, then alternative paths must be possible in the network being modelled; OD matrices should be time sliced so as to reproduce traffic demand dynamics correctly and the study time frame must extend beyond (earlier than) the peak hour to avoid starting the simulation in an oversaturated condition.

Calibration is an iterative process that consists of changing model parameters and comparing model outputs with a set of real data until a predefined level of agreement between the two data sets is achieved.

Which output needs to be compared depends on the type of model (macro, meso or micro), the objective of the study and the type of network. The most significant measures for a highway model are the relationship between speed/flow/density, lane utilization and congestion propagation. For an urban model, queue length, queue discharge speed and levels of services in large and/or more complex networks traffic flows and travel times become important as well.

It is important to emphasize that traffic counts are generally not sufficient for calibrations; as is well known (reference to fundamental equation), the same flow value can be reached in congested and uncongested conditions, so at least one more measure (speed, occupancy, etc.) is needed.

Once calibrated, the model must be validated comparing its outputs with a set of real data different from that used for calibration; if a predefined level of agreement between simulation and real-world data is achieved, the model can be considered valid, and thus suitable for studying future scenarios, subject to there being no changes to the real-world network that invalidate the model assumptions. The same considerations on output comparison made for calibration apply to validation as well.

For comparison purposes, Aimsun provides an interface capable of reading real data stored in ASCII files (space-, comma-, tab or semicolon-separated values), ODBC databases and GPX (GPS exchange format) files, linking them to model objects by id, external id or name.

5.2.3 Output Analysis

The dynamic (meso and micro) models produce time series in which each value is the aggregation of data collected during a regular interval defined by the user.

Mesoscopic simulations produce, for each section

- flow (also available for turnings)
- density
- speed
- travel time and delay time
- queue length

Microscopic simulations produce

- flow;
- density;
- speed and harmonic speed;
- travel time and delay time;
- queue length;
- stops and stop time;
- pollution and fuel consumption;
- trajectory data.

It is also possible to collect data at the level of ‘streams’ (a set of sections selected by the user) as well as being able to output global, network-wide values of the above outputs.

5.3 Fundamental Core Models: Car Following and Lane Changing

The core models in Aimsun deal with individual vehicles, each vehicle/driver having behavioural attributes assigned to them when they enter the system; those attributes remain constant during the whole trip. The difference between the core models at the mesoscopic and microscopic levels relates to the level of abstraction and to the process employed to update each vehicle's status. Accordingly, in what follows, we describe separately two sets of fundamental core models: microscopic behavioural models and mesoscopic behavioural models.

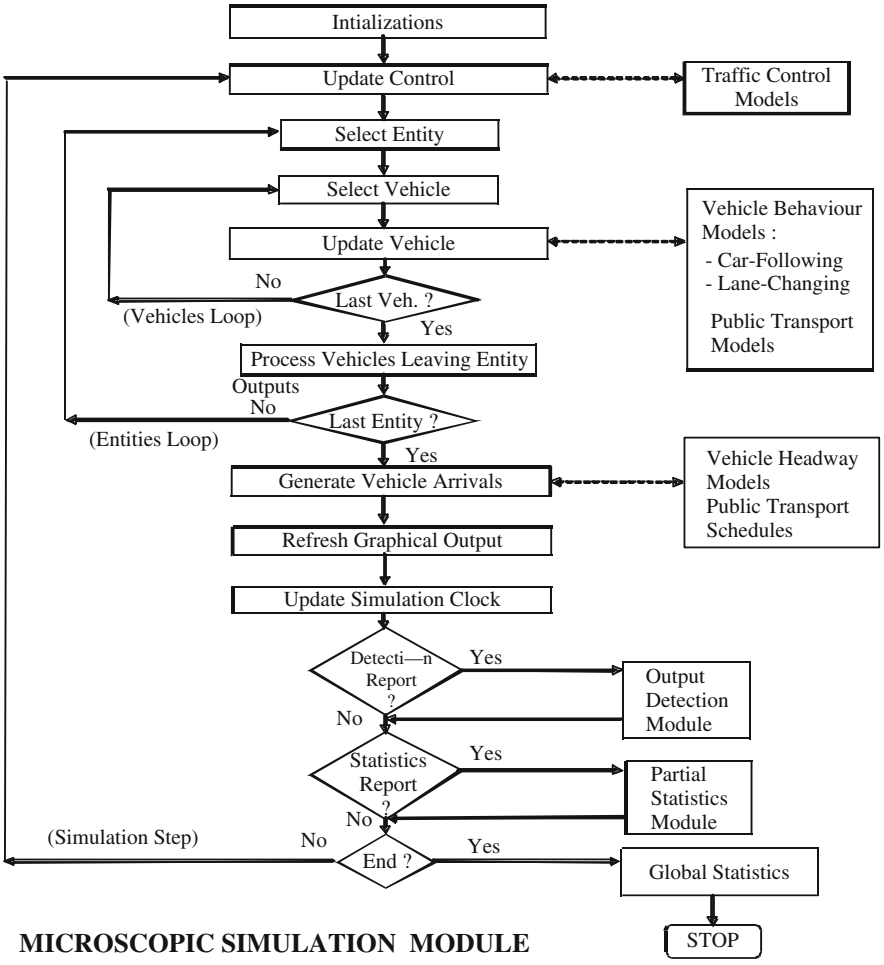


Fig. 5.1 The microscopic simulation process in Aimsun

5.3.1 Microscopic Logic of Simulation Process

The logic of the microscopic simulation process in Aimsun is illustrated in Fig. 5.1. It can be considered as a time slice-based simulation with an additional scheduled event calendar. At each time interval (simulation step), the simulation cycle updates the unconditional events scheduling list (i.e. events such as traffic light changes which do not depend on the termination of other activities). The 'Update Control' box in the flow chart represents this step. After this updating process, a set of nested loops starts to update the status of the entities (road sections and intersections) and vehicles in the model. Once the last entity has been updated, the simulator performs the remaining operations such as inputting new vehicles and collecting new data.

5.3.2 Mesoscopic Logic of simulation Process

The mesoscopic model in Aimsun works with individual vehicles but adopts a discrete-event simulation (Law and Kelton, 1991) approach, in which the simulation clock moves between events and there is no fixed time slice. An *event* is defined as an instantaneous occurrence that may change the state of the traffic network, i.e. the number of vehicles in sections and lanes, the status of the traffic signals etc. Events can be *scheduled* (known in advance to occur at a particular time in the simulation) or *conditional* (added to the event list dynamically during the simulation whenever some logical condition is satisfied). Specifically, a mesoscopic simulation includes the following types of events:

- Vehicle generation (vehicle entrance);
- Vehicle system entrance (virtual queue)
- Vehicle node movement (vehicle dynamics)
- Change in traffic light status (control)
- Statistics collection (outputs)
- Matrix change (traffic demand)

These events model the vehicle movements through sections and lanes by using a simplification of the car-following, lane-changing and gap-acceptance models used in the microscopic simulator. Nodes, on the other hand, are modelled as queue serves. All events have an associated time and a priority. Both these attributes are used to sort the event list. For example, events related to a change in the status of a traffic light or a new vehicle arrival are going to be treated before events that relate to statistics collection or vehicle movements inside a node.

5.3.3 Modelling Microscopic Vehicle Movement

In the Aimsun micro-simulator, during a vehicle's journey along the network, its position is updated according to two driver behaviour models termed 'car following' and 'lane changing'. The premise behind the models is that drivers tend to

travel at their desired speed in each road section but the environment (i.e. preceding vehicle, adjacent vehicles, traffic signals, signs, blockages, etc.) conditions their behaviour. Simulation time is split into small time intervals called simulation cycles or simulation steps (Δt).

At each simulation step, the position and speed of every vehicle in the system is updated according to the following algorithm:

if (it is necessary to change lanes) then
    Apply Lane-Changing Model
endif
if (the vehicle has not changed lanes) then
    Apply Car-Following Model
endif

Once all vehicles have been updated for the current simulation step, vehicles scheduled to arrive during this cycle are introduced into the system and the next vehicle arrival times are generated.

5.3.3.1 Microscopic Car Following

The car-following model implemented in Aimsun is based on the model proposed by Gipps (1981, 1986b). It can actually be considered an evolution of this empirical model, in which the model parameters are not global but determined by the influence of local parameters depending on the type of driver (speed limit acceptance of the vehicle), the road characteristics (speed limit on the section, speed limits on turnings, etc.), the influence of vehicles on adjacent lanes, etc.

The model consists of two components: acceleration and deceleration. The first represents the intention of a vehicle to achieve a certain desired speed, while the second reproduces the limitations imposed by the preceding vehicle when trying to drive at the desired speed.

This model states that the maximum speed to which a vehicle (n) can accelerate during a time period (t, $t+T$) is given as

$$V_a(n, t+T) = V(n,t) + 2.5a(n)T\left(1 - \frac{V(n,t)}{V^*(n)}\right)\sqrt{0.025 + \frac{V(n,t)}{V^*(n)}} \qquad (5.1)$$

where:

$V(n, t)$ is the speed of the vehicle n at time t;
$V^*(n)$ is the desired speed of the vehicle (n) for current position;
$a(n)$ is the maximum acceleration for the vehicle n;
T is the reaction time.

On the other hand, the maximum speed that the same vehicle (n) can reach during the same time interval (t, $t+T$), according to its own characteristics and the limitations imposed by the presence of the lead vehicle ($n-1$), is

$$V_b(n,t+T) = d(n)T + \sqrt{d(n)^2T^2 - d(n)\left[2\{x(n-1,t) - s(n-1) - x(n,t)\} - V(n,t)T - \frac{V(n-1,t)^2}{d'(n-1)}\right]} \tag{5.2}$$

where

$d(n)$ (< 0) is the maximum deceleration desired by vehicle n;
$x(n, t)$ is the position of the vehicle n at time t;
$x(n-1, t)$ is the position of the preceding vehicle $(n-1)$ at time t;
$s(n-1)$ is the effective length of the vehicle $(n-1)$;
$d'(n-1)$ is an estimation of the vehicle $(n-1)$ desired deceleration.

The speed of the vehicle (n) during time interval $(t, t+T)$ is the minimum of the two expressions above:

$$V(n, t+T) = \min\ \{\ V_a(n, t+T), V_b(n, t+T)\ \} \tag{5.3}$$

Then, the position of the vehicle n inside the current lane is updated taking this speed into the movement equation:

$$x(n, t+T) = x(n, t) + V(n, t+T)T \tag{5.4}$$

The car-following model is such that a leading vehicle, i.e. a vehicle driving freely without any vehicle affecting its behaviour, would try to drive at its maximum desired speed. Three parameters are used to calculate the maximum desired speed of a vehicle while driving on a particular section or turning; of those, two are related to the vehicle and one to the section or turning. Specifically

1. Maximum desired speed of the vehicle i: $v_{\max}(i)$
2. Speed acceptance of the vehicle i: $\theta(i)$
3. Speed limit of the section or turning s: $S_{\text{limit}}(s)$

The speed limit for a vehicle i on a section or a turning s, $s_{\text{limit}}(i, s)$, is calculated as follows:

$$s_{\text{limit}}(i, s) = S_{\text{limit}}(s) \cdot \theta(i) \tag{5.5}$$

Then, the maximum desired speed of the vehicle i on a section or a turning s, $v_{\max}(i, s)$, is calculated as follows:

$$v_{\max}(i, s) = MIN\,[s_{\text{limit}}(i, s), v_{\max}(i)] \tag{5.6}$$

This maximum desired speed $v_{\max}(i, s)$ is the same as that referred to above, in the Gipps' car-following model, as $V^*(n)$ [see eq. (5.1)].

The car-following model proposed by Gipps is a one-dimensional model that considers only the vehicle and its leader. However, the implementation of the car-following model in Aimsun also considers the influence of adjacent lanes. When a vehicle is driving along a section, we consider the influence that a certain number of vehicles driving slower in the adjacent right-side lane – or left-side lane when driving on the left – may have on the vehicle. The model determines a new maximum desired speed of a vehicle in the section, which will then be used in the car-following model, considering the mean speed of vehicles driving downstream of the vehicle in the adjacent slower lane and allowing a maximum difference of speed.

5.3.3.2 Microscopic Lane-Changing Model

The lane-changing model can also be considered as a development of the lane-changing model proposed by Gipps (1986a, b). Lane change is modelled as a decision process, analysing

- the desirability or necessity of a lane change (such as for imminent turning manoeuvres determined by the overall route that the vehicle is following);
- the benefits of a lane change (for example to reach the desired speed when the leading vehicle is slower); and
- the feasibility conditions for a lane change that are also local, depending on the location of the vehicle in the road network.

The lane-changing model is a decision model that approximates the driver's behaviour in the following manner: each time a vehicle has to be updated, we ask the following question*: Is it necessary or desirable to change lanes?* The answer to this question will depend on the distance to the next turning and the traffic conditions in the current lane. The traffic conditions are measured in terms of speed and queue lengths. When a driver is going slower than he wishes, he tries to overtake the preceding vehicle. On the other hand, when he is travelling fast enough, he tends to go back into the slower lane.

If we answer the previous question in affirmatively, to successfully change lanes, we must first answer two further questions:

- Is there benefit to changing lane? Check whether there will be any improvement in the traffic conditions for the driver as a result of lane changing. This improvement is measured in terms of speed and distance. If the speed in the future lane is fast enough compared to the current lane, or if the queue is short enough, then it is beneficial to change lanes.
- Is it feasible to change lanes? Verify that there is enough of a gap to make the lane change with complete safety. For this purpose, we calculate both the braking imposed by the future downstream vehicle to the changing vehicle and the braking imposed by the changing vehicle to the future upstream vehicle. If both braking ratios are acceptable, then the lane change is possible.

In order to achieve a more accurate representation of the driver's behaviour in the lane-changing decision process, three different zones inside a section are considered, each one corresponding to a different lane-changing motivation. These zones are characterized by the distance up to the end of the section, i.e., the next point of turning (see Fig. 5.2).

- *Zone 1*: This is the furthest distance from the next turning point. The lane-changing decisions are mainly governed by the traffic conditions of the lanes involved. The necessity of a future turning movement is not yet taken into account. To measure the improvement that the driver will get from changing lanes, we consider several parameters: desired speed of driver, speed and distance of current preceding vehicle, speed and distance of future preceding vehicle in the destination lane.
- *Zone 2*: This is the intermediate zone. It is mainly the desired turning lane that affects the lane-changing decision. Vehicles not driving in valid lanes (i.e. lanes where the desired turning movement can be made) tend to get closer to the correct side of the road from which the turn is allowed. Vehicles looking for a gap may try to adapt to it but do not yet affect the behaviour of vehicles in the adjacent lanes.
- *Zone 3*: This is the shortest distance to the next turning point. Vehicles are forced to reach their desired turning lanes, reducing speed if necessary, and even come to a complete stop (*gap forcing*) in order to make the change possible. Within this zone, vehicles in the adjacent lane may also modify their behaviour (*courtesy yielding*) in order to provide a gap big enough for the vehicle to succeed in changing lanes.

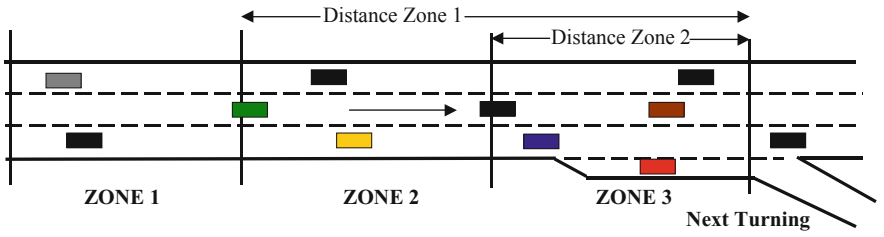


Fig. 5.2 Lane-changing zones

An overview of the lane-changing model is displayed in Fig. 5.3. The system identifies the type of entity (central lane, off-ramp lane, junction, on-ramp, etc.) into which the manoeuvre is to be carried out and then determines how zone modelling should be applied. The current traffic conditions are analysed, the level at which the lane change can be performed is determined and then the corresponding model is applied.

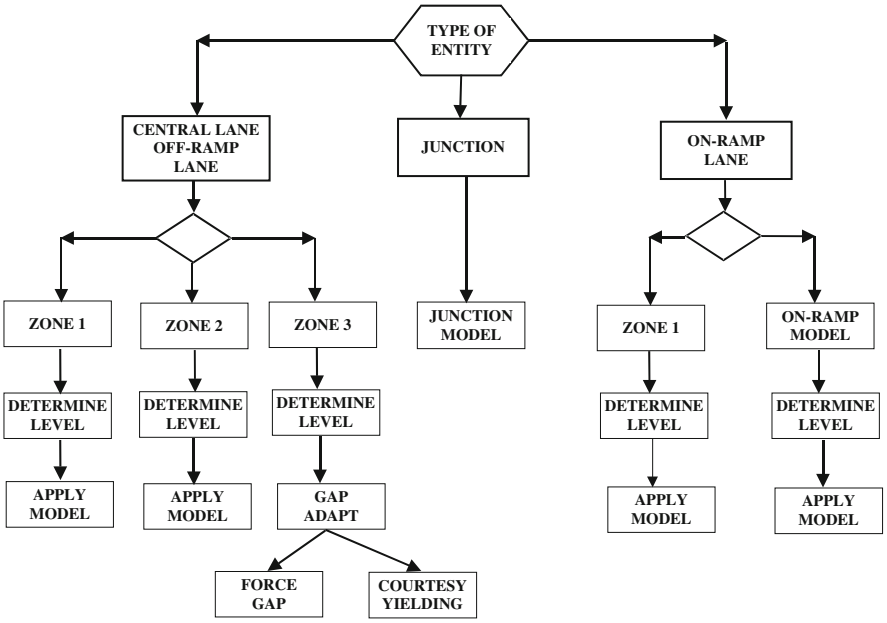


Fig. 5.3 Lane-changing model logic

5.3.3.3 Microscopic Look-Ahead Model

Until now, we have made the assumption that a vehicle driving along a section has knowledge only of its next turning movement, that is the turning it will take when arriving at the end of the current section. This means that the lane-changing decisions of each particular vehicle are made according to the next turning movement at the next intersection. However, this approach is not safe in urban networks with short sections or motorways/freeways with weaving sections that may be relatively short. In such situations and with very heavy traffic congestion, if we take into account only the next turning movement in the lane-changing decisions, it is possible that some vehicles will not reach the appropriate turning lane and consequently miss the next turn.

In order to avoid this undesirable behaviour as much as possible, Aimsun includes a look-ahead model, whose main purpose is to make the vehicles able to reach the turning lane in time. The idea is to provide vehicles with the knowledge of a set of next turning movements (defined by the user) based on which they can make lane-changing decisions.

The look-ahead model comprises four steps:

1. Knowing all sections belonging to the path that a vehicle follows, it considers the next turning movements ahead in the lane-changing behaviour. The lane-changing decisions are influenced by close lanes, the reserved lanes, bus stops,

in case of public transport vehicles, of all sections in its path and finally by the tuning movements ahead.
2. Lane-changing zones 2 and 3 of any section are extended back beyond the limits of the section, therefore affecting the upstream sections.
3. The next turning movement also influences the turning manoeuvres, so the selection of destination lane is also made based on the next turn.
4. Greater variability is given to the lane-changing zones in order to distribute the lane-changing manoeuvres along a longer distance.

5.3.3.4 Microscopic Gap-Acceptance Model

A gap-acceptance model is used to model give-way behaviour. This model determines whether a lower priority vehicle approaching an intersection can or cannot cross depending on the circumstances of higher priority vehicles (position and speed). This model takes into account the distance of vehicles to the hypothetical collision point, their speeds and their acceleration rates. It then determines the time needed by the vehicles to clear the intersection and produces a decision that is also a function of the level of risk of each driver.

Several vehicle parameters may influence the behaviour of the gap-acceptance model: acceleration rate, desired speed, speed acceptance and maximum give-way time. Other parameters, such as visibility distance at the intersection and turning speed, which are related to the section, may also have an effect. Among these, the acceleration rate, the maximum give-way time and the visibility distance at the intersection are the most important.

The acceleration rate gives the acceleration capability of the vehicle and therefore has a direct influence on the required safety gap. The maximum give-way time is used to determine when a driver starts to get impatient if he/she cannot identify a gap. When the driver has been waiting for more than this time, the safety margin (normally two simulation steps) is reduced to half of it (only one step).

The following algorithm is applied in order to determine whether a vehicle approaching a give-way sign can cross or not:

Given a vehicle (VEHY) approaching a give-way junction,

1. Obtain the closest higher priority vehicle (VEHP)
2. Determine the theoretical collision point (TCP)
3. Calculate time (TP1) needed by VEHY to reach TCP
4. Calculate estimated time (ETP1) needed by VEHP to reach TCP
5. Calculate time (TP2) needed by VEHY to cross TCP
6. Calculate estimated time (ETP2) needed by VEHP to clear the junction
7. If TP2 (plus a safety margin) is less than ETP1, vehicle VEHY will have enough time to cross; therefore it will accelerate and cross
8. Otherwise, if ETP2 (plus a safety margin) is less than TP1, vehicle VEHP would have already crossed TCP when VEHY had reached it; searching for the next closest vehicle with a higher priority, it would become VEHP and go to step 2
9. Else, vehicle VEHY must give way, decelerating and stopping if necessary

5.3.4 Modelling Mesoscopic Vehicle Movement

Mesoscopic vehicle movement in Aimsun is modelled depending on the location of a vehicle:

- Modelling vehicle movement in sections: car following and lane changing
- Modelling vehicle movement in nodes (node model):
  - Modelling vehicle movement in turnings
  - Modelling vehicle movement from sections to turnings: apply gap-acceptance model
  - Modelling vehicle movement from turnings to sections: apply lane selection model

Figure 5.4 illustrates mesoscopic vehicle movements in Aimsun.

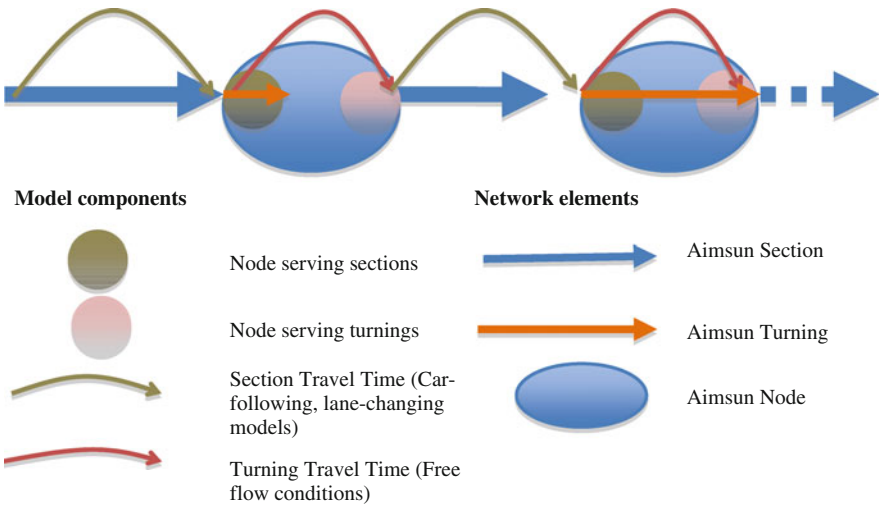


Fig. 5.4 Modelling vehicle movements

5.3.4.1 Mesoscopic Car following

In Aimsun, vehicles are assumed to move through sections and turnings, so sections and turnings are vehicle containers. That means that yellow boxes are not considered explicitly by the model. The section capacity, in terms of the number of vehicles that can stay at the same time in a section, is calculated by using the jam density (user-defined parameter) multiplied by the section length and the number of lanes. By

contrast, the turning capacity is calculated in a similar way, but using the feasible connections in the node instead of the number of lanes.

Car-following and lane-changing models are applied to calculate the section travel time. This is the earliest time a vehicle can reach the end of the section, taking into account the current status of the section (number of vehicles in the section). The modelling of vehicle movements inside sections in the Aimsun mesoscopic simulator is based on the work of Mahut (1999a, b, 2001).

Aimsun mesoscopic simulator is based on this node model that moves vehicles from one section to its next section of its path. This model contains two actions that take place in all nodes:

- *Serving sections*. This server calculates the next vehicle to enter the node. This is done by applying the gap-acceptance model and then using the exit times calculated by the car-following and lane-changing models mentioned above.
- *Serving turnings*. This server calculates the next vehicle to leave the node. The selection is done by applying the car-following and lane-changing models to calculate the travel time and getting the earliest time when a vehicle can enter in its downstream section.

5.3.4.2 Mesoscopic Lane Selection Model

This model is used to calculate the origin and destination lanes. These calculations are made during the treatment of the event called 'node event from turning', that is, before a vehicle enters into a section.

The mesoscopic simulator in Aimsun calculates a default movement from all exit lanes at the beginning of the simulation. This means that from each lane in a section there is a default next lane for each turning. Besides this default next lane choice, Aimsun mesoscopic is using two more heuristics in order to decide the next lane movement: obtaining the status of the next section and employing a look-ahead model.

From the default lane choice, the mesoscopic simulator looks for the best entrance lane from all turning destination lanes. By contrast, in comparable settings, the microscopic simulator looks only for a subset of all destination lanes.

In order to decide to change the default lane choice, Aimsun takes into account the density of all lanes and the cost of changing the default lane choice.

The other way to change the default lane choice is by applying the look-ahead model described early. The look-ahead model implemented in Aimsun is shared between the microscopic and mesoscopic simulators.

5.3.4.3 Mesoscopic Gap-Acceptance Model

The gap-acceptance model is used to model give-way behaviour. In particular, the model is used when resolving node events in order to decide which of two vehicles in a conflict movement has priority. The generic rule is a FIFO rule, except when

there is a traffic sign (such as stop or give-way signs) in which case a simplification of the gap-acceptance model described earlier is applied.

5.4 Dynamic Traffic Assignment

Traffic assignment is the process of determining how the traffic demand, usually defined in terms of an origin–destination matrix, is loaded onto a road network, providing the way to compute the traffic flows on the network links. The underlying hypothesis is that vehicles travel from origin to destinations in the network along the available routes connecting them. The characteristics of a traffic assignment procedure are determined by the hypothesis on how vehicles use the routes. The main modelling hypothesis is the concept of user equilibrium which states that vehicles try to minimize their individual travel times, that is, drivers choose the routes that they perceive as the shortest under the prevailing traffic conditions. This modelling hypothesis is formulated in terms of Wardrop's first principle: *The journey times on all the routes actually used are equal and less than those which would be experienced by a single vehicle on any unused route.*

Traffic assignment models based on this principle are known as user equilibrium models as opposed to models in which the objective is to optimize the total system travel time independently of individual preferences – for details, see either Sheffi (1985) or Florian and Hearn (1995).

The advent of intelligent transport systems (ITS) and more specifically advance traffic management systems (ATMS) and advanced traffic information systems (ATIS) has highlighted the requirement for models accounting for flow changes over time, that is dynamic models able to describe appropriately the time dependencies of traffic demand and the resulting traffic flows. The dynamic traffic assignment problem can thus be considered an extension of the traffic assignment problem described above, capable of describing how traffic flow patterns evolve in time and space on the network (Mahmassani, 2001). The approaches proposed to solve the DTA problem can be broadly classified into two classes: mathematical formulations looking for analytical solutions and simulation looking for approximate heuristic solutions. General simulation-based approaches (Tong and Wong, 2000; Lo and Szeto, 2002; Varia and Dhingra, 2004; Liu et al., 2005) explicitly or implicitly split the process into two parts: a route choice mechanism determining how the time-dependent flows are assigned onto the available paths at each time step and method to determine how these flows propagate in the network. A systematic approach based on these two components was proposed by Florian et al. (2001, 2002) (see Fig. 5.5).

Solving the DTA in Aimsun involves the conceptual diagram depicted in Fig. 5.5. In terms of the software, when the user selects the dynamic scenario dialog in Aimsun, the system offers two options – microscopic or mesoscopic, determining the simulation approach on which the network loading is based – and for each one two alternatives: DTA based on route choice models (Ben-Akiva and Bierlaire, 1999) or DTA based on DUE.

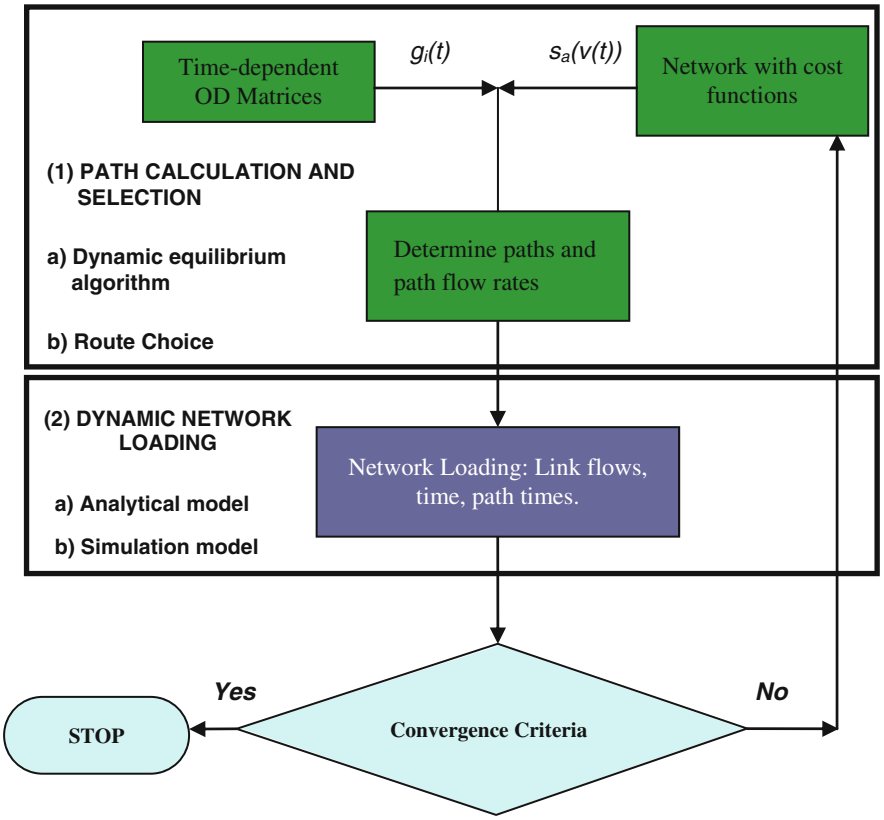


Fig. 5.5 Conceptual diagram of the heuristic dynamic assignment

The convergence criterion depends on the selected alternative: the completion of the demand loading in a one-off (single iteration) DTA based on route choice models and either the completion of the number of defined iterations or when the Rgap function reaches the desired accuracy in the DTA based on DUE.

An efficient computational implementation of this conceptual approach requires that the analytical part of the process, that is the path calculation and selection, is implemented independently of the dynamic network loading process selected to implement the heuristic part of the dynamic traffic assignment. In other words, assuming network consistency between the mesoscopic and microscopic representations, the path calculation based on time-dependent link costs must be the same; the only difference will lie in the values of the arguments of the link cost functions output by the mesoscopic or the microscopic traffic simulation.

Aimsun uses a common network representation, object model and database accessible by all models. In addition, both the microscopic and mesoscopic models are based on individual vehicles. This makes it possible to implement a 'dynamic traffic assignment server' (Barceló and Casas, 2006), whose conceptual structure

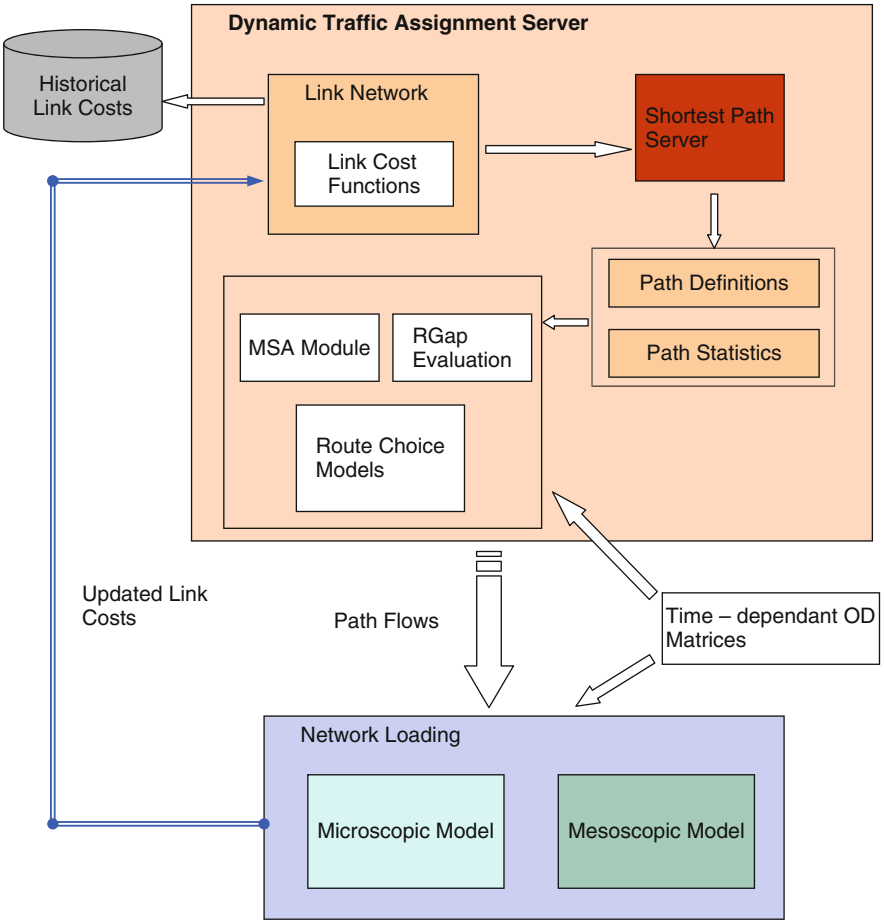


Fig. 5.6 Dynamic traffic assignment server structure

is depicted in Fig. 5.6. The dynamic traffic assignment server computes common shortest paths based on link cost functions evaluated in terms of current link costs or link cost accounting for stored values from previous iterations. Link costs are then updated by either a mesoscopic or a microscopic simulation approach. In what follows we describe three dynamic traffic assignment schemes that may be employed in a modelling study with Aimsun depending on the study objectives.

5.4.1 Dynamic Traffic Assignment Based on Discrete Choice Theory (Stochastic Route Choice)

Certain real-world situations, for example the reaction of drivers to a non-recurring incident, create a requirement for dynamic traffic assignment mechanisms whose

output is not necessarily optimal in the sense of the dynamic version of Wardrop's principle (Friesz et al., 1993; Smith, 1993; Ran and Boyce, 1996). In such cases the route choice mechanism tries to optimize route selection decisions based on the currently available information. One such mechanism that accounts for uncertainties in the information available to drivers is the use of choice that includes the possibility of en route rerouting mechanisms, based either on discrete choice theory or on other probabilistic approaches (Mahmassani, 2001). These approaches provide solutions to the dynamic traffic assignment problem that does not seek dynamic user equilibrium (DUE).

One of the DTA methods in Aimsun is based on stochastic route choice (Barceló and Casas, 2002, 2004a, b, 2006).

The simulation process based on time-dependent routes consists of the following steps:

Repeat until all the demand has been assigned:

1. Calculate initial shortest routes for each OD pair using the defined initial costs.
2. Simulate for a user-defined time interval assigning to the available routes the fraction of the trips between each OD pair for that time interval according to the selected route choice model and obtaining new average link travel times as a result of the simulation.
3. Recalculate shortest routes, taking into account the current average link travel times.
4. If there are guided vehicles, or variable message signs suggesting rerouting, provide the information calculated in step 3 to the drivers that are dynamically allowed to reroute.
5. Go to step 2.

In the proposed network loading mechanism based on microscopic or mesoscopic simulation, vehicles follow paths from their origins in the network to their destinations. So the first step in the simulation process is to assign a path to each vehicle when it enters the network, from its origin to its destination. This assignment, made by a path selection process based on a discrete route choice model, will determine the path flow rates.

Given a finite set of alternative paths, the path selection calculates the probability of each available path and then the driver's decision is modelled by randomly selecting an alternative path according to the probabilities assigned to each alternative. Route choice functions represent implicitly a model of user behaviour that emulates the most likely criteria employed by drivers to decide between alternative routes in terms of the user's perceived utility (or, more precisely, a disutility or cost in the case of trip decisions) defined in terms of perceived travel times, route lengths, expected traffic conditions along the route, etc.

The logit, c-logit and proportional route choice functions are the default route choice functions available in Aimsun. Additional choice functions may be introduced by the user, using Aimsun's function editor.

The multinomial logit route choice model defines the choice probability P_k of alternative path k, $k \in K_i$, as a function of the difference between the measured utilities of that path and all other alternative paths:

$$P_k = \frac{e^{\theta V_k}}{\sum_{j \in K_i} e^{\theta V_j}} = \frac{1}{1 + \sum_{j \neq k} e^{\theta (V_j - V_k)}} \tag{5.7}$$

where V_j is the perceived utility for alternative path j (i.e. the opposite of the path cost or path travel time) and θ is a scale factor that plays a twofold role: on the one hand, it makes the decision based on differences between utilities independent of measurement units on the other hand, it influences the standard error of the distribution of expected utilities, determining in that way a trend towards utilizing many alternative routes or concentrating on very few routes. In that sense, θ is the critical parameter to calibrate so that the logit route choice model leads to a meaningful selection of routes.

A drawback in using the logit function is a tendency towards route oscillations in the routes used, with the corresponding instability creating a kind of 'flip-flop' process. According to our experience, there are two main reasons for this behaviour: the properties of the logit function and the inability of the logit function to distinguish between two alternative routes when there is a high degree of overlapping.

The instability of the routes used can be substantially improved when the network topology allows for alternative routes with little or no overlapping at all, playing with the shape factor of the logit function and re-computing the routes very frequently. However, in large networks where many alternative routes between origins and destinations exist, and some of them exhibit a certain degree of overlapping, the use of the logit function may still exhibit some weaknesses. To avoid this drawback, the c-logit model, proposed in Cascetta et al. (1996) and Ben-Akiva and Bierlaire (1999), is also available as an option. In this model, the choice probability P_k, of each alternative path k belonging to the set K_i of available paths connecting the ith OD pair, is defined as

$$P_k = \frac{e^{\theta (V_k - \mathrm{CF}_k)}}{\sum_{j \in K_i} e^{\theta (V_j - \mathrm{CF}_j)}} \tag{5.8}$$

where V_j is the perceived utility for alternative path j, i.e. the opposite of the path cost, and θ is the scale factor, as in the case of the logit model. The term CF_k, denoted as 'commonality factor' of path k, is directly proportional to the degree of overlapping of path k with other alternative paths. Thus, highly overlapping paths have a larger CF factor and therefore smaller utility compared to paths with a similar perceived utility and less overlapping. CF_k is calculated as follows:

$$\mathrm{CF}_k = \beta \cdot \ln \sum_{j \in K_i} \left(\frac{L_{jk}}{L_j^{1/2} L_k^{1/2}} \right)^{\gamma} \tag{5.9}$$

where L_{jk} is the length of arcs common to paths j and k, while L_j and L_k are the length of paths j and k, respectively. Depending on the two factor parameters β and γ, a greater or a lesser weighting is given to the 'commonality factor'. Larger values of β means that the overlapping factor has greater importance with respect to the utility V_j; γ is a positive parameter, whose influence is smaller than β and which has the opposite effect. The utility V_j used in this model for path j is the opposite of the path travel time tt_j (or path cost depending on how it has been defined by the user).

Another option is the estimation of the choice probability P_k of path k, $k \in K_i$, in terms of a generalization of Kirchhoff's laws given by the function

$$P_k = \frac{\mathrm{CP}_k^{-\alpha}}{\sum_{j \in K_i} \mathrm{CP}_j^{-\alpha}} \tag{5.10}$$

where CP_j is the cost of the path j, α is in this case the parameter whose value has to be calibrated.

5.4.2 *Dynamic Traffic Assignment via an Iterative Heuristic (Stochastic Route Choice with Memory/Additional Information)*

The formulation described in this section is useful for scenarios in which ATMS and ATIS applications transmit reliable (simulation-based) traffic forecasts to drivers, allowing them to adjust their behaviour accordingly. Alternatively it may be considered as a model of the process by which travellers adjust their current information with conjectures about the expected traffic conditions ahead; this could correspond to the process followed by commuters adapting their behaviour according to a day-to-day learning process depending on the fluctuations of traffic patterns until they consider that no further improvement is possible. The implementation in Aimsun combines dynamic (mesoscopic or microscopic) simulations with an iterative heuristic procedure that mimics the day-to-day learning process that attempts to reach DUE, though with no guarantee of convergence (Barceló and Casas, 2002, 2006; Liu et al., 2005).

The iterative heuristics replicates the simulation N times and link costs for each link j, for each time interval t, $t+1$, ..., L (where $L = T/\Delta t$, T being the simulation horizon and Δt the user-defined time interval in which to update paths and path flows) at every iteration n stored. Thus at iteration n, the link costs of previous iteration $n-1$ can be used as an anticipatory mechanism to estimate the expected link cost at the current iteration. Let $s_a^{jl}(v)$ be the current cost of link a with flow v at iteration l of replication j. Then the average link costs for the future $L-l$ time intervals, based on the experienced link costs for the previous $j-1$ replications, is given as

$$\bar{s}_a^{j,l+i}(v) = \frac{1}{j-1}\sum_{m=1}^{j-1} s_a^{m,l+i}(v); i = 1, ..., L-l \tag{5.11}$$

The 'forecasted' link cost can then be computed as

$$\widetilde{s}_a^{j,l+1}(v) = \sum_{i=0}^{L-l} \alpha_i \bar{s}_a^{j,l+i}(v); \quad \text{where } \sum_{i=0}^{L-l} \alpha_i = 1,\ \alpha_i \geq 0,\ \forall i \text{ are weighting factors} \tag{5.12}$$

The resulting cost of path k for the ith OD pair is given as

$$\widetilde{S}_k(h^{l+1}) = \sum_{a \in A} \widetilde{s}_a^{j,l+1}(v)\, \delta_{ak} \tag{5.13}$$

where, usually, δ_{ak}, the arc-path incidence matrix, is 1 if link a belongs to path k and 0 otherwise. The path costs $\widetilde{S}_k\left(h^{l+1}\right)$ are the arguments of the route choice function (logit, c-logit, proportional or user-defined) used at iteration l+1 to distribute the demand g_i^{l+1} across the available paths for OD pair i.

The default implementation in the current version of Aimsun uses a simplified version consisting of the link cost function:

$$c_{it}^{k+1} = \lambda c_{it}^{k} + (1-\lambda)\,\widetilde{c}_{it}^{k} \tag{5.14}$$

where c_{it}^{k+1} is the cost of using link i at time t at iteration k+1 and c_{it}^{k} and $\widetilde{c}_{it}^{k}$ correspond to the expected and experienced link costs, respectively, at this time interval from previous iterations.

5.4.3 Dynamic Traffic Assignment via the Method of Successive Averages (Dynamic User Equilibrium)

Several road infrastructure modification programmes give rise to the problem of understanding 'steady-state' time-dependent path choice after the modifications have been in place for a while. In many cases, the motivation at planning level is not to understand the pattern and duration of the transition from the current state to the final state (something attempted by the iterative heuristics described in the previous section) but rather to approximate that final state assuming that Wardrop's generalized principle applies (i.e. reaching or approaching dynamic user equilibrium).

The method of successive averages (MSA) procedure redistributes the flows among the available paths in an iterative procedure that at iteration n computes a new shortest path from origin r to destination s at time interval t, $c_{rs}(t)$; the path flow update process is as follows:

Case a $c_{rs}(t) \notin P_{rs}^{n}(t)$

$$f_{rsp}^{n+1}(t) = \begin{cases} \alpha_n f_{rsp}^{n}(t) & \text{if } p \in P_{rs}^{n}(t) \\ & \forall r, s, t \\ (1-\alpha_n)\, d_{rs}(t) & \text{if } p = c_{rs}(t) \end{cases}$$

Let $P_{rs}^{n+1}(t) = P_{rs}^{n}(t) \cup c_{rs}(t)$
Case b $c_{rs}(t) \in P_{rs}^{n}(t)$

$$f_{rsp}^{n+1}(t) = \begin{cases} \alpha_n f_{rsp}^{n}(t), & \text{if } p \neq c_{rs}(t) \\ & \forall r, s, t \\ \alpha_n f_{rsp}^{n+1}(t) + (1-\alpha_n)\, d_{rs}(t), & \text{if } p = c_{rs}(t) \end{cases}$$

Let $P_{rs}^{n+1}(t) = P_{rs}^{n}(t)$
where $f_{rsp}(t)$ is the flow on path p from r to s departing origin r at time interval t, $P_{rs}(t)$ is the set of all available paths from r to s at time interval t and $d_{rs(t)}$ is the demand (number of trips) from r to s at time interval t.

Depending on the values of the weighting coefficients α_n, different MSA schemes can be implemented (Carey and Ge, 2007), perhaps the most typical value is $\alpha_n = n/(n+1)$.

One of the potential computational drawbacks of these implementations of MSA is the growing number of paths in the case of large networks; to avoid this in the case of the DTA assignments in Aimsun, the user has the option to specify the maximum number K of paths to keep for each origin–destination pair. Therefore in implementing the MSA in Aimsun, it was considered that it would be desirable to keep this feature.

Several modified MSA implementations have been proposed to keep control on the number of paths in MSA algorithms (Peeta and Mahmassani, 1995; Sbayti et al., 2007); however, possibly one of the most efficient computationally is the one proposed by Florian et al. (2002) modifying the MSA algorithm to keep bounded the number of alternative paths to account for each origin–destination pair. This variant of the algorithm initializes the process on the basis of an incremental loading scheme distributing the demand among the available shortest paths; the process is repeated for a predetermined number of iterations after which no new paths are added and the corresponding fraction of the demand is redistributed according to the MSA scheme. The modified MSA works as follows:

Let K be the maximum number of iterations to compute new paths.
Case a: $n \leq K$: a new shortest path $c_{rs}(t) \notin P_{rs}^{n}(t)$ is found

$$f_{rsp}^{n+1}(t) = \frac{1}{n+1} d_{rs}(t),\ \forall p \in P_{rs}^{n}(t), \forall (r,s) \in \Im, \forall t \in [0,T]$$

Let $P_{rs}^{n+1}(t) = P_{rs}^{n}(t) \cup c_{rs}(t)$

Case b: $n > K$: the new shortest path is computed among the existing paths $c_{rs}(t) \in P_{rs}$, and the set P_{rs} does not change

$$f_{rsp}^{n+1}(t) = \begin{cases} \frac{n}{n+1} f_{rsp}^{n}(t) & \text{if } p \neq c_{rs}(t) \\ \frac{n}{n+1} f_{rsp}^{n+1}(t) + \frac{1}{n+1} d_{rs}(t) & \text{if } p = c_{rs}(t) \end{cases} \quad \forall p \in P_{rs}, \forall (r,s) \in \Im, \forall t \in [0,T]$$

This is the version of the MSA algorithm implemented in Aimsun. However, taking into account the possibility of repeating shortest paths from one iteration to next to keep a maximum of K different shortest paths, a proper implementation of the algorithm requires that the number of iterations n is defined by OD pair and time interval.

All the proposed approaches for DUE are based on simulation procedures for the network loading process and therefore are heuristic in nature; therefore no formal proof of convergence can be provided, and consequently a way of empirically determining whether the solution reached can be interpreted in terms of a DUE, in the sense that 'the actual travel time experienced by travellers departing at the same time are equal and minimal', can be based on an ad hoc version of the relative gap function proposed by Janson (1991):

$$\text{Rgap}(n) = \frac{\sum_{t} \sum_{(r,s)\in\Im} \sum_{p\in P_{rs}(t)} f_{rsp}^{n}(t) \left[\tau_{rsp}^{n} - \theta_{rs}^{n}(t)\right]}{\sum_{t} \sum_{(r,s)\in\Im} d_{rs}(t)\, \theta_{rs}^{n}(t)} \tag{5.15}$$

where $f_{rsp}^{n}(t)$ is the flow on path p from r to s departing origin r at time t at iteration n and the difference $\tau_{rsp}^{n}(t) - \theta_{rs}^{n}(t)$ measures the excess cost experienced by the fact of using a path of cost $\tau_{rsp}^{n}(t)$ instead of the shortest path of cost $\theta_{rs}^{n}(t)$ at iteration n. The ratio measures the total excess cost with respect to the total minimum cost if all travellers had used shortest paths.

5.4.4 Methodology and Data Flows for Dynamic Traffic Assignment

The introductory section has highlighted the advantages of full integration whereby all models share a network representation, modelling object data and demand data. From a practitioner's standpoint what is crucially important is the ability to combine model outputs, thus giving rise to sophisticated workflows where models are applied sequentially, iteratively or even concurrently (Barceló and Casas, 2006). Figure 5.7 illustrates the possibilities that open up in terms of workflows.

We distinguish two main sets of data flows:

- OD matrix data flows
- Path assignment data flows

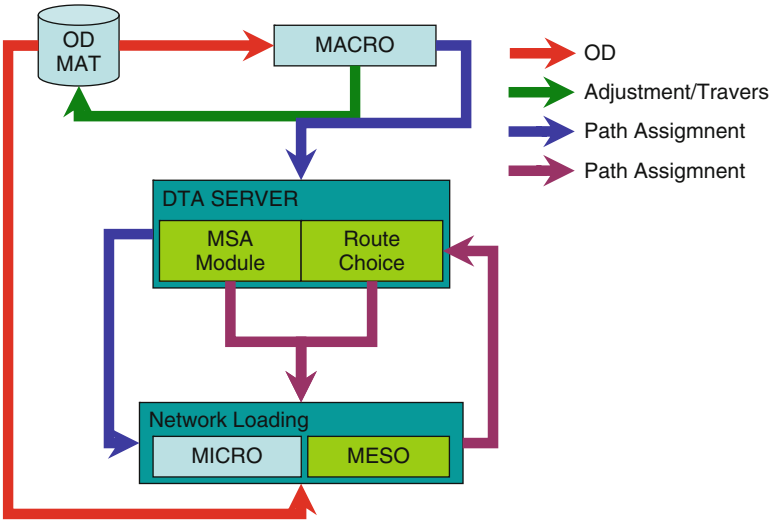


Fig. 5.7 Macro–meso–micro data flows

5.4.4.1 OD Matrix Data Flows

- To refine the inputs for the microscopic/mesoscopic simulation:
  - Estimating demand for future scenarios by means of growing factor analysis and matrix-balancing procedures.
  - Adjusting the global OD matrix from the available traffic counts on a subset of links.
- To start the analysis at the macrolevel for a large urban or metropolitan area and refine the analysis at the micro/mesolevel conducting detailed dynamic simulation experiments of selected subareas:
  - Defining interactively the window spanning the selected subarea
  - Calculating the corresponding traversal matrix
  - Adjusting the traversal from traffic counts of links in the network spanning the subarea
  - Running the simulation experiments for the subarea model

5.4.4.2 Path Assignment Data Flows

The main path assignment data flows considering the functional architecture and the integration of macro–meso–micro could be summarized as follows:

- Path assignment results could be the output of the following:
  - *Static traffic assignment applying the macroscopic model*: The path definition and the path flow rates are per the whole simulation time (no per time period).

The result, as a consequence of the static user equilibrium, could be interpreted as a reflection of a recurrent situation or historical situation without the concept of time.
- *Dynamic traffic assignment applying the DUE approach with independency of the type of network loading process (either could be mesoscopic or microscopic model)*: In that case, the path definition and the path flow rates are per time interval (defined by the user). The result, as a consequence of the dynamic user equilibrium, could be interpreted as a reflection of a recurrent situation over the time, giving the 'normal' traffic behaviour or profile, considering the time dimension.

- Path assignment results could be the input of the following:
  - *Dynamic traffic assignment applying the dynamic user equilibrium approach*: The path assignment result, concretely the path definition, could be used as initial paths, instead to start with the paths calculated in free-flow conditions, per time slice if the path assignment result comes from a dynamic traffic assignment based on DUE approach using either mesoscopic or microscopic model as network loading process or per the whole simulation if the path assignment result comes from a static traffic assignment using the macroscopic model.
  - *Dynamic traffic assignment based on discrete route choice models*: The path assignment result of either a static traffic assignment or a dynamic traffic assignment based on DUE, interpreted as the recurrent situation, could be used to define the paths and the path flow rates for a subset of the drivers (for instance, defining a certain percentage of use of the path assignment result) and the rest of the vehicles decide the path according to the dynamic traffic assignment based on discrete route choice models.

The different alternatives of generating the path assignment results depending on the type of traffic assignment (either static or dynamic) could be interpreted:

1. the user equilibrium interpreted as the result of the recurrent or day-to-day learning processes of the drivers, either dynamic or static, and
2. the dynamic traffic assignment based on discrete route choice models interpreted as the vehicles receive information about the current situation.

We can consider different applications of the path assignment result data flow in projects:

- Model the addition of a new infrastructure or the modification of a current infrastructure:
  - One possibility is the evaluation of the scenario base (the current situation) and the future scenario, applying the dynamic traffic assignment based on dynamic user equilibrium in order to model the recurrent situation in both scenarios.

  - The other possibility is the evaluation of the previous scenarios, but including an intermediate scenario (transitory scenario) that represents the temporal modifications of the infrastructure due to the road works (such as lane closures, speed reductions, rerouting actions). In this intermediate scenario the recurrent situation probably is not representative, because there is no day-to-day learning process and the behaviour of the drivers could be helped by traffic management policies. In that case, the most appropriate approach is to combine the day-to-day learning process of the current situation, which means the use of the path assignment result of a dynamic user equilibrium, with a dynamic traffic assignment based on discrete route choice models in order to model the transition and the effect of the temporal traffic management policies.

- Analyse microscopically or mesoscopically a subarea but considering the influence in a wide area:
  - Simulate the wide area using the dynamic traffic assignment based on dynamic user equilibrium with the mesoscopic model as network loading.
  - The path assignment result of the previous simulation could be used as input for a simulation combined with a dynamic traffic assignment based on discrete route choice models in order to model the changes of behaviour in the function of the new traffic management or the design of the subarea. The level of detail of the subarea determines the type of the network loading to consider: microscopic or mesoscopic, independently of the traffic assignment approach.

5.5 Calibration and Validation of Aimsun models

5.5.1 General Remarks

Validation is the process of testing that a model represents a viable and useful alternative means to real experimentation. This requires the exercise of *calibrating the model,* that is adjusting model parameters until the resulting output data agree closely with the system observed data. The validation of the simulation model will be established by comparing the observed output data from the actual system and the output data from the simulation experiments conducted with the computer model.

Model calibration and validation is inherently a statistical process in which the uncertainty due to data and model errors should be accounted for. Depending on the variables selected, the system and simulated data available, and their characteristics and statistical behaviour, there exists a variety of statistical techniques either for paired comparisons or for multiple comparisons and time series analysis. The conceptual framework for this validation methodology is described in the diagram of Fig. 5.8. According to this logic, when the results of the comparison analysis are not acceptable to the degree of significance defined by the analyst, the rejection of the

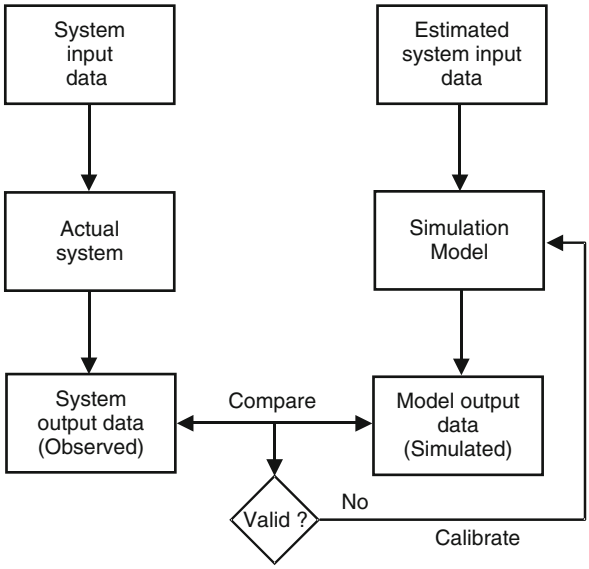


Fig. 5.8 Conceptual framework for model validation

simulation results implies the need for recalibrating some aspects of the simulation model. The process is repeated until a statistically significant degree of similitude is achieved.

In the case of traffic systems, the behaviour of the actual system is usually defined in terms of flows, speeds, occupancies, queue lengths, and so on, which can be measured by traffic detectors at specific locations in the road network. To validate the traffic simulation model, Aimsun emulates the traffic detection process and produces a series of simulated observations whose comparison to the actual measurements is used to determine whether the desired accuracy in reproducing the system behaviour is achieved. Rouphail and Sacks (2003) propose the following set of guiding principles:

1. The analyst must be aware that calibration and validation are conducted in particular contexts.
2. Depending on the context, the model requires specific sets of relevant data.
3. Both models and field data contain uncertainties.
4. Feedback is necessary for model use and development.
5. Model validation must be exercised on an independent data set from the calibration data set.

The analyst will have to identify which data are relevant for the planned study, collect them, identify the uncertainties, filter out the data accordingly and use two independent sets of data. *The first set should be used for calibrating the model parameters and the second for running the calibrated model and then for validating the calibrated model.*

The key question in the diagram of figure 5.8: "Is the model valid?" can then be reformulated as, "Do model results represent faithfully the aspect of reality that is material to the study?" The statistical techniques provide a quantified answer to this question, quantification that, according to Rouphail and Sacks, can be formally stated in the following terms: the probability that the difference between the 'reality' and the simulated output is less than a specified tolerable difference within a given level of significance:

$$P\{\ |\text{'reality'} - \text{simulated output}|\ \leq d\ \} > \alpha$$

where d is the tolerable difference threshold indicating how close the model is to reality and α is the level of significance that tells the analyst how certain is the result achieved.

5.5.2 Verification and Validation in Aimsun

5.5.2.1 Verification

The main components of a traffic simulation model are the following:

1. The geometric representation of the road traffic network and related objects (traffic detectors, variable message panels, traffic lights, etc.)
2. The representation of traffic management schemes (directions of vehicle's movement, allowed and banned turnings, etc.), and of traffic control schemes (phasing, timings, offsets)
3. The individual vehicle behavioural models: car following, lane change, gap acceptance, etc.
4. The representation of the traffic demand
5. Input flow patterns at input sections to the road model and turning percentages at intersections
6. Time-sliced OD matrices for each vehicle class
7. The dynamic traffic assignment model

The graphical edition in Aimsun has been designed with the objective of supporting the user in tasks (1) and (2) of the process of building the road network model. To facilitate these tasks, Aimsun accepts as a background a digital map of the road network, in terms of a DXF file from a GIS or an AutoCAD system, a JPEG or a bitmap file, etc.; so sections and nodes can be built subsequently into the foreground. Aimsun supports both urban and interurban roads, which means that the level of detail covers elements such as surface roads, entrance and exit ramps, intersections, traffic lights and ramp metering.

The use of the graphic editors on the digital maps of the road networks provides the basis for a continuous visual validation of the quality of the geometric model. At

the same time, the auxiliary online debugging tools in Aimsun prevent the most blatant mistakes in building the geometric representation, warning the modeller when obvious inconsistencies may occur.

In other words, the Aimsun model-building process is assisted with validation tools to check the correctness of the geometric model of the road network within the limits of logic rules. Some aspects may lie beyond the analysis capabilities of the assistance software, i.e. whether banning a turning is correct or not. This decision may depend on the objectives of the traffic management scheme defined by the traffic manager. Something similar could be said regarding whether or not to include a movement in a phase. However, a different case might be that of a previously defined movement that was not included in any phase; this is something that can be checked by the assistance tools.

In this way, Aimsun ensures a geometric model exhibiting a 'high face validity' that could even be further validated by the modeller through visual inspection facilitated by the graphic display of the Aimsun model.

In order to make the validation of geometry easier, Aimsun offers two tools for checking whether there are errors in the network definition or not and also gives facilities for fixing these errors.

In Fig. 5.9 an example of a network with a centroid configuration is shown. The network checker detected three problems:

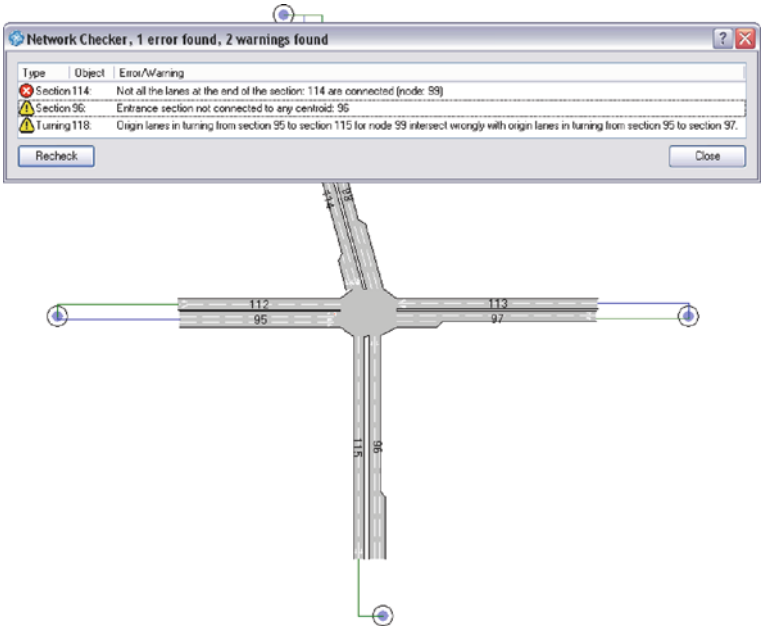


Fig. 5.9 Checking the network including centroids

- At the end of section 114, the lane on the right has no turning defined; this can be easily observed, as the node does not cover this lane.
- Section 96 is an entrance section (vehicles should enter the network through it) but it is not connected to an origin centroid generating traffic.
- There are two turnings, from section 95 to 115 and from section 95 to section 97, which would make the vehicles to collide.

Another tool available for the verification phase is the dynamic network checker (see Fig. 5.10). The purpose of this component is to detect problems within a running simulation. The specific features are the following:

- Count lost vehicles in nodes: Track all vehicles that have been unable to make their desired turning in a node.
- Stationary vehicle detection: Determine the vehicle that has been stationary for a time greater than *user-defined parameter.* Any of three actions may be taken. (i) The vehicle may be recorded in the log (either the log window or the file defined as *Output File*), (ii) the simulation may be stopped by selecting *Stop Simulation* and (iii) the vehicle may be automatically removed by selecting *Automatically Remove Vehicle*. The check may be applied only to specific sections of interest (and reduce execution time) by specifying section ids in a comma-separated list in the field *Apply to Sections*.

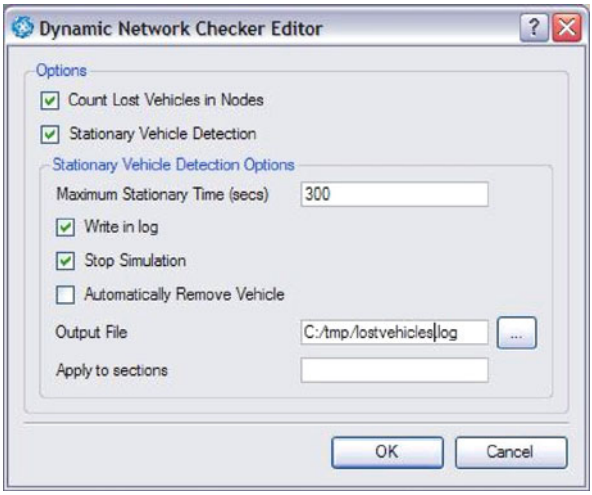


Fig. 5.10 Dynamic network checker editor

5.5.3 Validation

The statistical methods and techniques for validating simulation models are clearly explained in most textbooks and specialized papers (Law and Kelton, 1991; Balci,

1998; Kleijnen, 1995) and the validation process is independent of the traffic simulation model.

From the general methodology, three main principles that establish a framework for model validation are (Barceló and Casas, 2004b) as follows:

1. The measured data in the actual system should be split into two data sets: the data set that will be used to develop and calibrate the model and a separate data set that will be used for the validation test.
2. The data collection process is specified in the system as well as in the simulation model: traffic variables or MOEs (i.e. flows, occupancies, speeds, service levels, travel times, etc.), whose values will be collected to be used for the calibration and validation phases, and the collection frequency (i.e. 30 s, 1 min, 5 min, etc.).
3. According to the methodological diagrams in Fig. 2, validation should be considered an iterative process; at each step in the iterative validation process, a simulation experiment will be conducted. Each of these simulation experiments will be defined by the data input to the simulation model, the set of values of the model parameters that identify the experiment and the sampling interval.

The validation process based on standard statistical comparison between model and system outputs (Barceló and Casas, 2004b) could be summarized as follows:

- Comparison based on global measurements
- Comparison based on time series analysis
- Comparison based on band analysis

5.5.3.1 Comparison Based on Global Measurements

A method that has been widely used in validating transport-planning models, in the typical situation in which only aggregated values are available (i.e. flow counts at detection stations aggregated to the hour), has been to analyse the scattergram or, alternatively, to use a global indicator as the GEH index, widely used in practice in the United Kingdom. Figure 5.11 depicts an example of such analysis. The regression line of observed versus simulated flows at 76 detection stations for the aggregated 1 h values is plotted along with the 95% prediction interval. The R^2 value of 93.4 and the fact that only three points lay out of the confidence band would lead to the conclusion that the model could be accepted as significantly close to the reality.

For the same example the GEH value is 72%; therefore the model would have been rejected, leading to a conclusion contradicting the previous one.

Independent of the considerations of whether one criterion is better that the other, one should draw the attention that this type of indicators can be considered only as a primary indicator for acceptance or rejection in the case of microscopic simulation models. As indicators working with aggregated values, they do not capture what

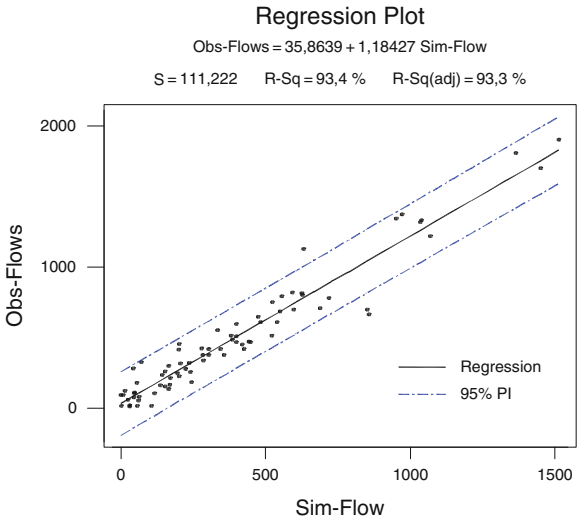


Fig. 5.11 Example of scattergram analysis to compare observed versus simulated aggregated flows

is considered the essence of the microscopic traffic simulation: the ability to capture the time variability of traffic phenomena. Therefore other types of statistical comparison should be proposed.

5.5.3.2 Comparison Based on Time Series Analysis

Theil's U-statistic (Theil, 1966) is the measure achieving the above-mentioned objectives of overcoming the drawbacks of the RMSE index and taking into account explicitly the fact that we are comparing two autocorrelated time series, and therefore the objective of the comparison is to determine how close both time series are.

An immediate interpretation of Theil's U-statistic is the following:

$U = 0 \Leftrightarrow$ the forecast is perfect
$U = 1 \Leftrightarrow$ the forecast is as bad as possible

Figure 5.12 depicts the statistics provided by Aimsun for helping in the validation of the model:

- Regression analysis
- Theil's coefficient

This information could also be displayed over the space as a global view in the network, considering the GEH index or the Theil's coefficient (see Fig. 5.13)

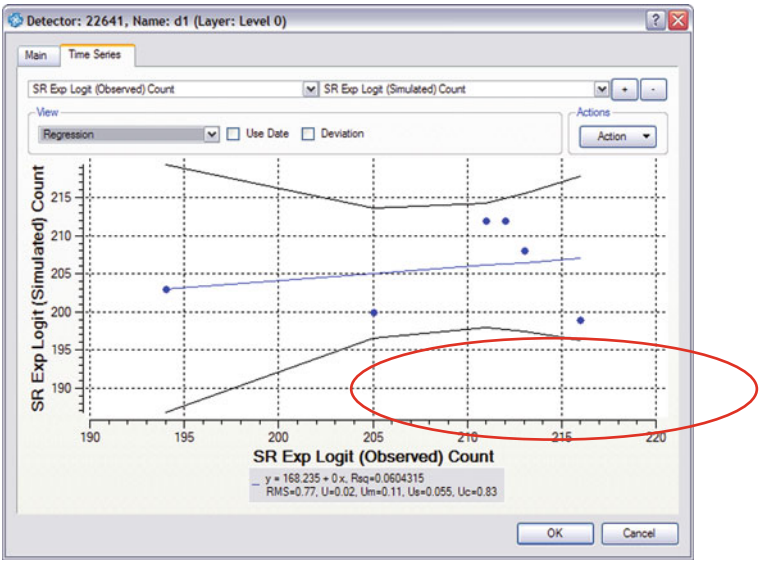


Fig. 5.12 Statistics for model validation

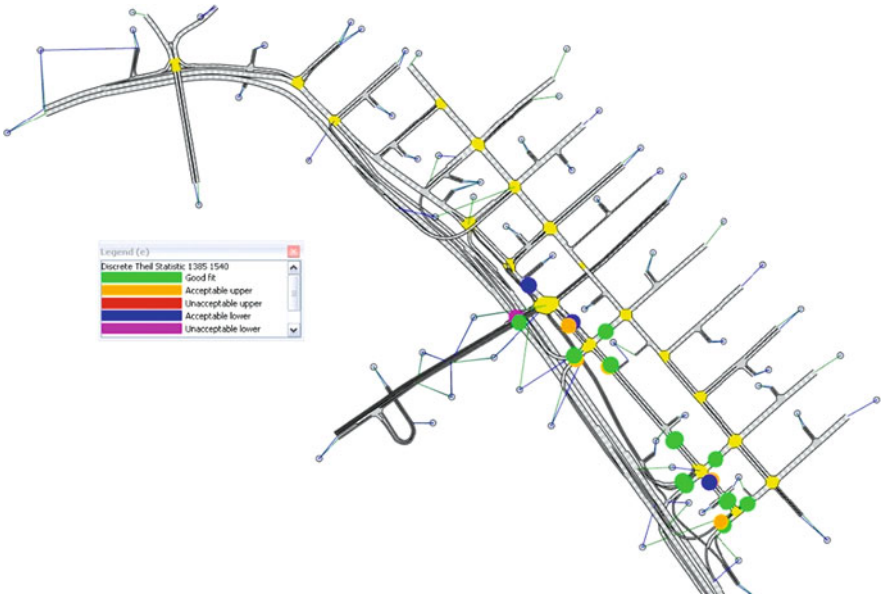


Fig. 5.13 Theil's validation global view

5.5.3.3 Comparison Based on Band Analysis

When the data collection can be automated and traffic data can be collected for long time periods (i.e. flow counts for *n* Mondays for rush hour from 7:00 until 9:00 am), the comparison between the measured data and the simulated data could consist of the comparison of two bundles of time series, the set of measured time series and the set of time series resulting from independent replications of the simulation model. Validation could then be based on developing suitable statistical procedures to compare (see Fig. 5.14) the following:

- single/mean pattern to single/mean pattern
- mean pattern to bandwidth
- bandwidth to bandwidth

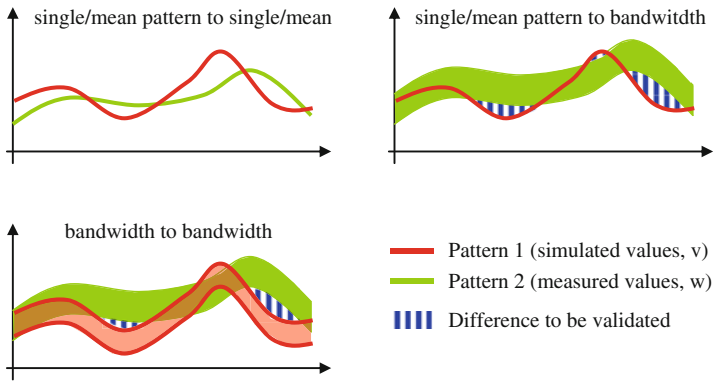


Fig. 5.14 Possibilities of comparison

5.5.4 Calibration

In the case of a traffic simulation model, model behaviour depends on a rich variety of model parameters. The model is composed of entities, i.e. vehicles, sections, junctions, intersections, and so on, each of them described by a set of attributes, i.e. parameters of the car following, the lane changing, gap acceptance, speed limits and speed acceptance on sections, and so on; the model behaviour is determined by the numerical values of these parameters. The calibration process has the objective of finding the values of these parameters that will produce a valid model. *Model parameters must be supplied with values. Calibration is the process of obtaining such values from field data in a particular setting and this process is completely dependent on the simulation model.* Some examples will help to illustrate this dependency between parameter values and model behaviour. Vehicle lengths have a clear influence on flows: as the vehicle lengths increase, flows decrease and queue lengths

increase. In the typical car-following models, the target speed, the section speed limit and the speed acceptance, among others, define the desired speed for each vehicle on each section. The higher the target speed, the higher the desired speed for any given section, resulting in an increase in flow according to the flow–speed relationships. In this way, as part of the calibration process, one should establish for a particular model the influence of acceleration and breaking parameters on the capacity of the sections, namely for weaving sections. Similarly, depending on how the lane changing is modelled, the effects of lengths of zones where the lane-changing decision can be made influence the capacity of the weaving sections, especially when these lengths are local parameters whose values could depend on traffic condition. These effects also happen when parameters influencing the lane distribution are included in the model. Table 5.2 is an example of the influence of microscopic parameters.

The calibration process proposed in Aimsun has relationship with the type of parameter (behavioural model or dynamic traffic assignment model) and their nature (global parameters and local parameters). This process stresses the calibration of the global parameters in front of the local parameters in order to avoid the risk of entering in a overcalibration situation that deals with a situation where it is not possible to extrapolate the results obtained in a future scenario where the local calibration will

Table 5.2 Influence of micro parameters

Level	Parameter	Influence
Vehicle	Maximum desired speed Normal and maximum deceleration Maximum acceleration Speed acceptance	Speed, travel time, queue discharge, lane changing, etc.
	Minimum distance	Capacity, queue length
	Give-way time	Yield and on-ramp capacity, lane-changing blockages
	Guidance acceptance	Use of new routes
	Reaction time	Section capacity, on-ramp capacity
	Reaction time at stop	Stop and go capacity, queue measures
Global	Queue up and leaving speeds	Queue statistics
	Two lanes car following	Smoothing traffic, merging situations
	Lane-changing parameters	Distribution among lanes, interurban situations
	Speed limit	Average speed, travel times
	Turning speed	Turning capacity, travel times, average speed
Section	Visibility distance	Yield sign behaviour
	Distance zones	Turning proportions, blockings
	Distance on-ramp	On-ramp capacity, use of slow lane
	Yellow box speed	Junction capacity

be not possible because of the lack of real data. This process could be summarized as follows:

1. Calibration of behavioural models using global parameters (all vehicle type parameters, such as reaction time, reaction time at stop, speed acceptance, etc.)
2. Calibration of behavioural models using local parameters (all section and node parameters that have an influence on the vehicle behaviour, such as local reaction time variation, jam density, lane-changing zonification of the section)
3. Calibration of dynamic traffic assignment using global parameters (number of different alternatives to consider, the time interval, the default cost functions parameters, etc.)
4. Calibration of dynamic traffic assignment using local parameters (scale factor per OD pair, cost function for an individual section, etc.)

5.5.4.1 Calibration of Behavioural Models

Obviously the most exact procedure to calibrate behavioural model parameters (such as car-following model, lane-changing model and gap-acceptance model) is to conduct specific experiments in which accurate field data are recorded on the relative distances and speeds between pairs of leader–follower vehicles, and the simulation model is calibrated against the field data. A recent example of these types of experiments can be found in Manstetten et al. (1998) and Bleile et al. (1996). Unfortunately these types of experiments are expensive and can seldom be conducted in the current professional practice, but recently, inside the NGSIM project (http://www.ngsim.fhwa.dot.gov), the availability of the trajectory data in different topologies (freeways, arterial, merging, etc.) allows this type of calibration. Figure 5.15 shows the type of analysis to be conducted with respect to the speed and gap between the leader and the follower.

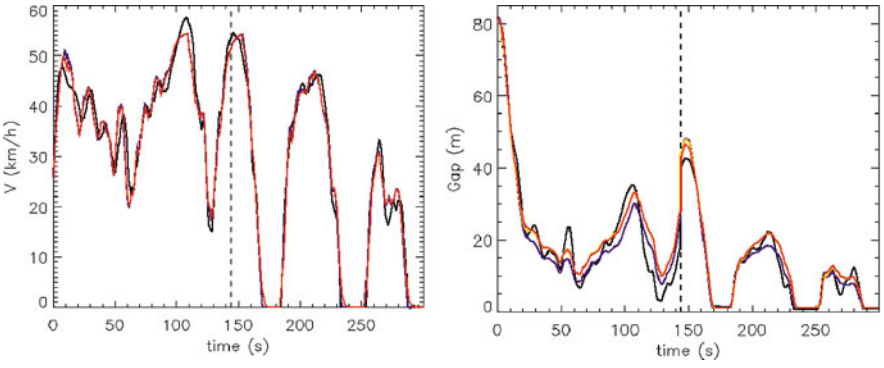


Fig. 5.15 Speed and gap profiles. Simulated versus observed

However, taking into account that correct car-following and lane-changing models acceptably calibrated must be capable of reproducing accurately enough macroscopic observable phenomena, as, for example, flow–occupancy or flow–density relationships, additional tests to analyse the quality of the microscopic simulator can be conducted to check the ability to reproduce such macroscopic behaviour. Manstetten et al. (1998) propose a test based on simulating increasing flows on a closed ring model, as the one depicted in Fig. 5.16, to reproduce a priori estimated flow–density relationships. A steadily increasing flow is injected in

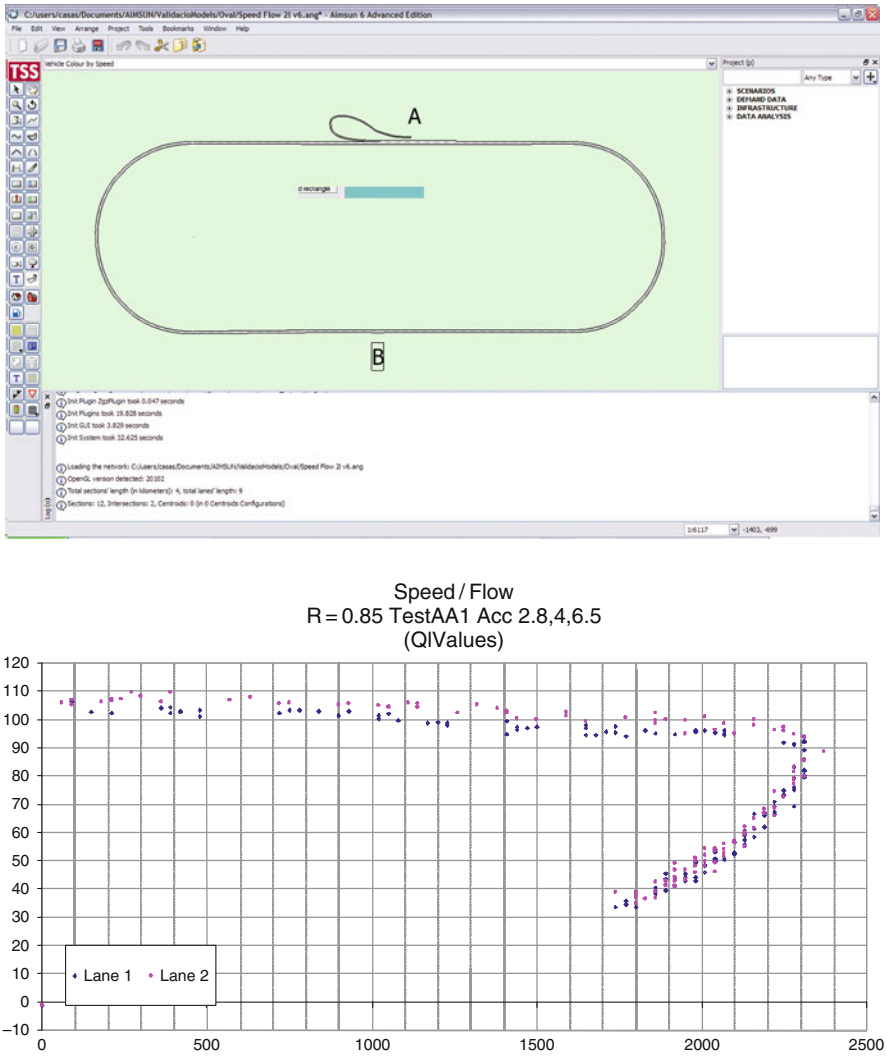


Fig. 5.16 Model to estimate speed–flow curves and example

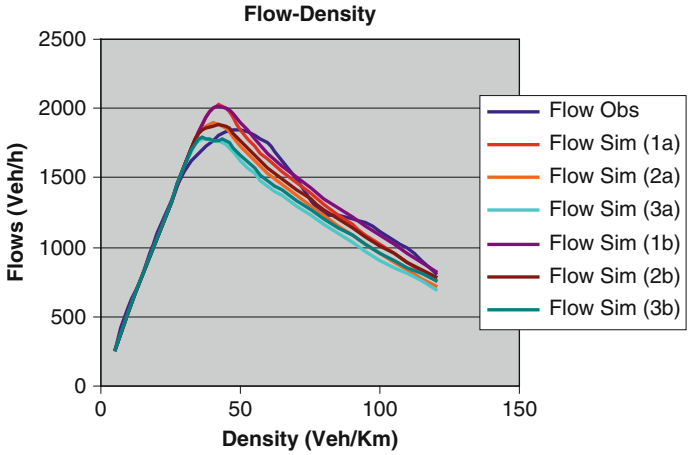


Fig. 5.17 Empirical versus simulated flow–density curves

the model at on-ramp A until reaching saturation after a predefined time horizon. A detector at B collects the traffic data. Figure 5.16 also displays the speed–flow graphics for a reaction time of RT=0.85 s, acceleration normally distributed with mean acceleration 2.8 m/s^2, standard deviation 0.56 m/s^2, normal deceleration 4.0 m/s^2 and maximum deceleration 6.0 m/s^2, providing a capacity of 2,320 vphpl.

The results of Aimsun for the simulated flow–density curve versus the empirical one for this test are displayed in Fig. 5.17, and they appear to be fairly reasonable as confirmed by the values of the RMS error measuring the fitting between the measured and the simulated values. The graphics in Fig. 5.17 also shows the sensitivity of the Aimsun car-following model to variations in the values of two model parameters, the reaction time and the minimum vehicle-to-vehicle distance (effective length). A subset of the simulation experiments to determine the values of the model parameters best fitting the observed values is summarized in Fig. 5.17, showing that the best fitting is achieved in the simulation experiment 1b with a reaction time of 0.9 s and an effective length equal to the vehicle length plus 0.75 m (Table 5.3).

Model parameters: reaction time (RT, s) and effective vehicle length (vehicle length+DM, m)

Table 5.3 Model quality as a function of reaction time and effective vehicle length

	Simulation 1a	Simulation 2a	Simulation 3a	Simulation 1b	Simulation 2b	Simulation 3b
	RT0.9/ DM1.0	RT0.95/ DM1.0	RT1.0/ DM1.0	RT0.90/ DM0.75	RT0.95/ DM0.75	RT1.0/ DM0.75
RMS	0.0645901	0.091316	0.121131	0.0518984	0.0620237	0.0920621

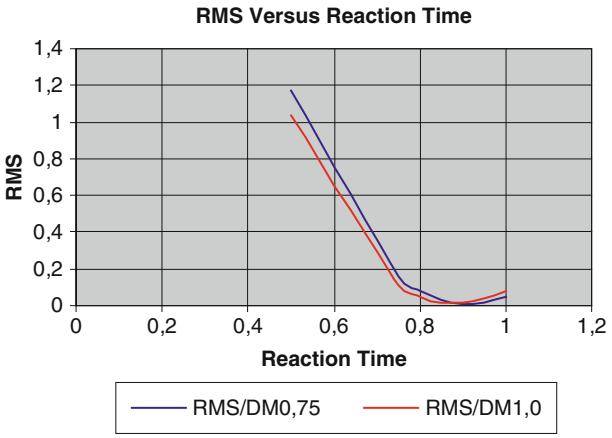


Fig. 5.18 RMS versus reaction time

The graphics in Fig. 5.18 displays the variation of the RMS error as a function of the reaction time parameter in the car-following model as implemented in AIMSUN; the blue curve corresponds to a fixed value of the minimum distance between vehicles of 0.75 m and the red one to 1 m. This illustrates in more detail the example of pilot runs of the simulation model to calibrate the parameters of the car-following model for a given context.

Similar simple models to understand and adjust the parameter values to fit the situations to study have been proposed by Yoshii (1999).

5.5.4.2 Calibration of Dynamic Traffic Assignment

Dynamic traffic assignment calibration is performed comparing traffic flows, possibly disaggregated by turning, and travel times. In urban networks the turning flows are limited by the signals (it can be roughly estimated calculating green/cycle ratio). A manual check of the reasonableness of alternative paths built between OD pairs can also be useful to detect cost errors.

An example of calibration of the dynamic traffic assignment are the computational results conducted with networks of various types and sizes (Barceló and Casas, 2006); Figure 5.19a depicts the time evolution of the Rgap function for the logit route choice function, and Fig. 5.19b depicts the plot of the Rgap versus GEH index of all replications using the logit route choice model for the network of the city of Preston in the UK which has 415 links (road sections), 165 intersections and 33 origin–destination centroids. The cloud of points that are in the area of the acceptable Rgap and GEH index represents 70% of the experiments in which the logit route choice was used. The cloud of points that are in the area of the acceptable Rgap and GEH index represents 70% of the experiments in which the logit route choice was used.

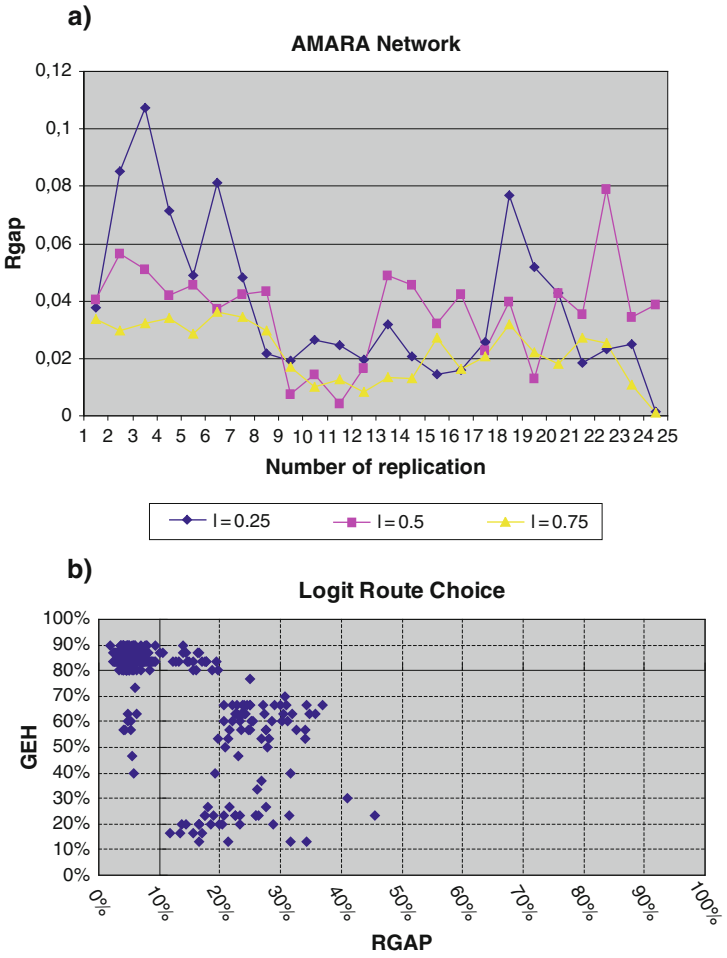
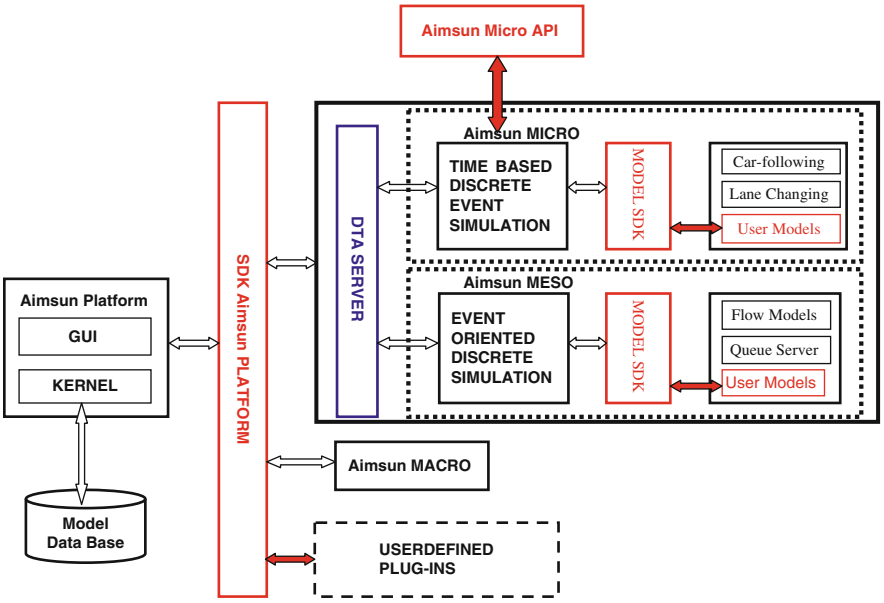


Fig. 5.19 (**a**) Rgap validation; (**b**) Rgap and GEH validation

5.6 Extended Modelling Capabilities: Working with External Applications

The functional architecture of Aimsun, shown in Fig. 5.20, allows the user different extended modelling capabilities, each one with a different role and objectives for working with external applications. The different possibilities, represented by red arrows and text boxes, are as follows:

- SDK Aimsun platform
- Micro API (APPI)
- Micro/mesomodel SDK



- SDK Aimsun Platform
- Micro API (APPI)
- Micro/Meso Model SDK

Fig. 5.20 Functional architecture of Aimsun

5.6.1 SDK Aimsun Platform

The SDK (software development kit) Aimsun Platform is a collection of libraries, header files, documents and samples that allow any user or company to develop applications for, or based on, Aimsun. This tool allows the interaction between an external application (in Fig. 5.20, user-defined plug-ins) and the Aimsun platform, which is divided into two main parts: (1) the kernel that contains all objects and their definition that are part of the application domain and (2) the graphical user interface (GUI) that contains all objects needed to implement/modify the user interface (such as dialogs, drawers and controls).

Aimsun application domain is the transportation domain. All the systems have been designed to support transportation-related applications (such as traffic simulators, location problems and assignment models). This specialization of the model offers to the developer facilities not found on other, more general, systems such as streets/roads, OD matrices, control plans and topology information. The user could develop and connect external applications allowing the interaction at level of the model definition and/or the graphical user interface. The developing language is either C++ or Python.

Examples of the use of this tool are ALMO, a software for detection analysis and traffic pattern recognition (www.momatec.de), and Legion, a pedestrian simulator (www.legion.com)

5.6.2 Micro API

The Aimsun micro API (application programme interface) is a tool or a module that gives Aimsun the ability to interface with almost any *external application* that may need access to some objects of Aimsun micro-simulator during simulation run time. The development language could be either C, C++ or Python.

This tool defines a set of functions that allow to get information from any object during the simulation time (such as vehicle information, detector measurements, statistical data, network information, demand information and traffic control plan information) and set information (such as change vehicle attributes, network object attributes, demand attributes and traffic control plan attributes). The exchange of information between the external application and the Aimsun micro-simulator is done in every simulation step.

Different applications of the micro API could be the following:

- *Traffic control/management system as external application*: The micro-simulator emulates the detection process. Then, through a set of functions, it provides the external application with the required *simulation detection data* (e.g. flow and occupancy). The external application uses this data to feed some control policy and decides which control and/or management actions have to be applied on the road network. Finally, the external application sends the corresponding actuations (e.g. change the traffic signal state, the phase duration and display a message in a VMS) to the simulation model, which then emulates their operation through the corresponding model components such as traffic signals, VMS' and ramp metering signs. The more representative examples could be the connection with the following external applications: SCATS (www.rta.nsw.gov.au), VS-PLUS (www.vs-plus.com), SCOOT (www.siemens.co.uk), UTOPIA (www.miz.it) and TUC (www.dssl.tuc.gr).
- *Vehicle-simulated data as external application*: Another uses of the Aimsun API are to access detailed vehicle-simulated data to feed some user's model (e.g. fuel consumption and pollution emissions), to keep track of a guided vehicle throughout the network by an external vehicle guidance system and to simulate the activities of vehicles such as floating cars.
- *Vehicle driving as external application*: Another uses of the Aimsun API are to access detailed vehicle-simulated data and drive a subset of the vehicles by the external application. For example, as external application, a driving simulator drives where Aimsun creates the more realistic 3D scenario for the driving simulator, according to the traffic flow theory. One example of this use is SCANeR (www.scaner2.com).

5.6.3 Micro/Mesomodel SDK

The dynamic models in Aimsun (microscopic and mesoscopic) are based on behavioural models, such as car following and lane changing, provided by default by Aimsun. The micro/mesomodel SDK is a tool that allows the implementation of new behavioural models and then during the simulation overwrites the default behavioural models. In other words, this tool could be considered as an API specifically oriented to functionalities to implement the behavioural models, such as get the leader's vehicle, get the vehicle upstream and downstream in a target lane and set a new positions and speed.

Probably the most representative example is the use of the tool to include and evaluate in Aimsun the different behavioural models developed inside the NGSIM project scope (sponsored by the FHWA).

5.7 Selected Overview of Advanced Case Studies and Applications

In this chapter we present three different applications of Aimsun to transportation engineering problems. The first two case studies focus on the use of microsimulation. The third one demonstrates the need for a combined use of the macro and micromodel. As such, it serves to illustrate the benefit of having several models integrated in the same software application.

5.7.1 The Paris Tramway

In order to improve the public transport of Paris, the French capital authorities decided to put in place a new tramway line whose itinerary will follow the so-called Boulevards des Maréchaux urban ring road. The study discussed here focussed mainly on the eastern area of the boulevards and particularly on the segment between Porte de Vincennes and Porte de Bagnolet. This tramway line will be physically separated from the boulevards with the exception of intersections. Urban planners considered four design scenarios, namely axial, bilateral and two variants of the latter.

In order to maximize modal shift to this new public facility, trams should offer competitive travel times, which means that they should not stop on signalized intersections. Consequently, tram pre-emption systems had to be designed whilst respecting that the fixed control plans is used in the rest of the city's intersections. This pre-emption system having a notable impact on the vicinity road network, the Paris City Council commissioned a study aiming to compare the four design scenarios based on the following criteria, ranked in priority order:

- Pedestrian safety
- Tramway speed

- Capacity of the crossing streets
- Capacity of the boulevards

Analytical approaches are better suited to the study of isolated intersections and are less well suited to evaluating the impacts of the pre-emption system applied on the 15 intersections of the study area. Therefore, the Paris City Council decided to undertake a traffic simulation study using the Aimsun software, and in particular the micro-simulator.

As already mentioned, the control plan of an intersection was to remain fixed while no tram was near the intersection. A fully adaptive control plan was ruled out. Thus, when a tram call is received, the current phase has to be ended (respecting the minimum green time), an inter-phase has to be activated and finally the special tram phase must be set off. Note that this tram phase has to be activated prior to the tram reaching the braking area before the intersection. Once the tram exits the intersection (exit call) and when no other tram is at the intersection, the control plan activates an inter-phase making it possible to return to the next phase of the fixed plan.

In the first phase of the study, we simulated a single intersection considered as representative (Fig. 5.21). For each design scenario, we tested several control plan options. Following the general objectives put forward by the Paris City Council, our assessment identified, for each scenario, the best control plan that would be applied to the whole network.

In addition, we analysed separately the performance of two signalized roundabouts included in the study area. The determination of the correct phases and timings was challenging: gridlocks or situations with vehicles blocking the tramway had to be carefully studied in order to get the best timings.

The final step in the analysis consisted of micro-simulations of the whole tram corridor to assess the global performance of each design and associated optimized pre-emption scheme. The first output used was the tram speed profiles in order to test if there was any deceleration other than that induced by scheduled tram stops. This allowed detecting any problem of vehicles getting trapped on the tramway or any sub-optimalities of the pre-emption system. Capacity, or better said, queue length increases have then been measured at each point in the network in order to identify bottlenecks and to evaluate the risk of congestion propagation that could lead to an intersection blockage. For each intersection, we compared the upstream demands to the downstream throughput to get the total queue increase (in number of vehicles per hour). Finally, we analysed safety aspects in terms of numbers of potential conflicts between pedestrians, bicycles (both were included in the simulation model) and cars. In addition to that, non-quantitative aspects were highlighted, such as the probability of red light violations, thanks to the input of experienced local engineers who were able to identify situations that favour such violations.

The global evaluation, based on a multicriteria approach, finally showed that the axial scenario achieves the best performance. Therefore, this design was selected for the implementation phase of the Tramway des Maréchaux (east side). The revised

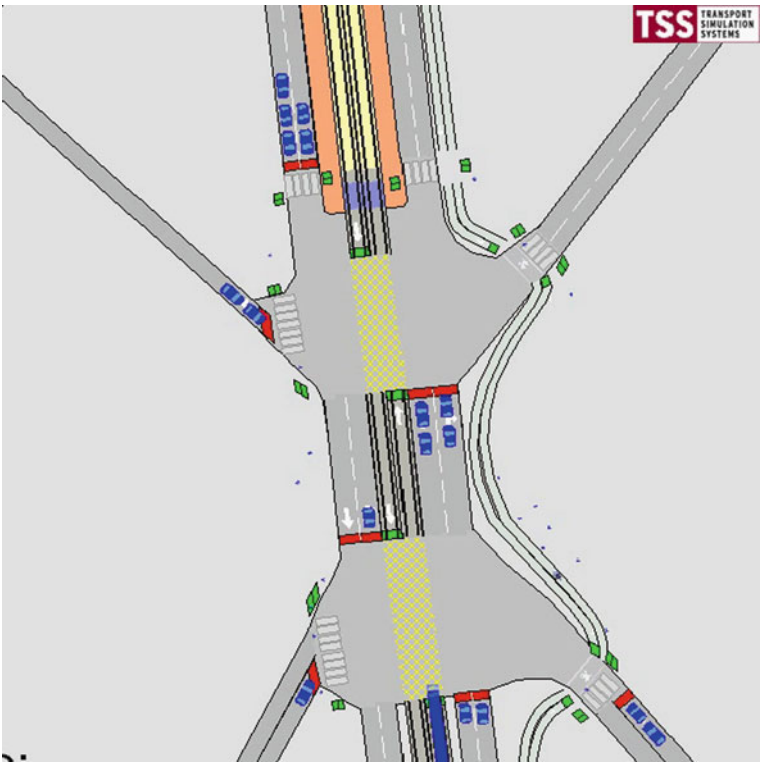


Fig. 5.21 Snapshot of a tram priority-designed intersection micro-simulation

signal control plans, including tram pre-emption, that we designed for this simulation study will be used for real implementation after some refinement related to practical limitations.

5.7.2 Behaviour-Based Highway Performance Assessment

In this section we provide an overview of a study commissioned by the Royal Automobile Club of Catalonia (RACC). The study focussed on the influence of driver lane-changing and lane usage behaviours on highway performance in the area of Catalonia.

Anecdotal evidence suggests that current driver behaviour on Catalonian highways does not follow the rules that should theoretically apply. The issue is not restricted to speed limit violations; inadequate lane changes can also be observed. The overuse of the left ('fast') lane on two-lane highways and of the central lane for three-lane highways is a frequently observed phenomenon. However, little information is available on the positive or negative implications of such behaviour on

capacity, performance and safety levels. The aim of our study was to shed light on those relationships.

Forcing real drivers to behave in different ways in order to compare several types of driver behaviour in the same highway segment would be a costly, impractical and potentially unsafe undertaking. Micro-simulation in Aimsun was unsurprisingly seen as a safer, faster and much more cost-efficient approach.

To enable a quantitative assessment of highway capacity and safety levels depending on different lane-changing behaviours, we devised the following three scenarios:

- Base scenario (current situation)
- 'Legal' scenario
- 'No-rules' scenario

The first scenario reflects the behaviour that can currently be observed on Catalonian highways. The second one aims to model a situation in which driver behaviour is in strict compliance with the highway code and can be summarized as: 'Recovering the rightmost available lane when not overtaking and using the fast lane only during overtaking'. Finally, the third scenario can be thought of as the opposite extreme of the second one. No restrictions on lane usage apply, which means that drivers can use the lane they feel better on – as well as having the option of undertaking slower cars.

Using a segment of three-lane highway as a test bed, we analysed the impact of risky overtaking on flows and in particular on its potential to create congestion. To better understand the dynamics of this phenomenon of 'spontaneous congestion creation', we recorded and analysed 3D videos of the simulation.

The calibration of the Aimsun simulation parameters, based on historical traffic data set from that same three-lane highway segment, enabled us to reproduce current driver behaviour to a very high level of fidelity. The characteristic overuse of the central lane was reproduced particularly well. The data we used for calibration included distribution curves of flows per lane and flow-versus-speed data for each lane (Fig. 5.22).

Results from the simulations of all three scenarios showed that the base scenario (with 7,700 veh/h) achieves a slightly reduced capacity compared to the other two (8,000 veh/h). However, this difference being limited, an analysis using other highway sections and data sets should be done to confirm this tendency.

From a safety point of view, our analysis focussed on speed differences between lanes. The key premise of our approach was that higher values of speed difference imply lower safety levels. The idea is that speed difference increases collision probability when changing lane. The simulation results showed clearly that the 'least safe' scenario is the base one. By contrast, the 'no-rules' scenario turned out to be the safest configuration. Specifically, the latter scenario corresponded to the most homogeneous speed distribution in the traffic flow.

To measure performance, we undertook a comparison of average speeds. Our objective was to identify which scenario offers the lowest average travel time to road

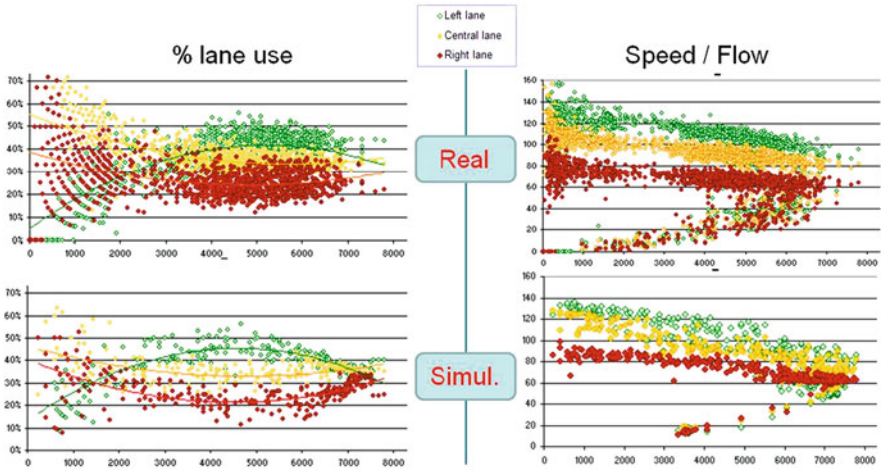


Fig. 5.22 Validation of the percentage of lane use and the speed–flow relationship for different highway traffic volumes (field data at the *top* and simulated data at the *bottom*)

users. Simulation outputs suggest that the 'legal' scenario is the best performing when this indicator is used. By contrast, the base scenario had the worst performance in this respect.

In conclusion, the current patterns of highway lane changing and lane use behaviour in Catalonia give rise to a very slight sacrifice on capacity compared to the two alternative scenarios we analysed. However, in what concerns safety and travel time, the alternative scenarios outperform substantially the existing situation. To conclude, the 'no-rules' and 'legal' approaches are better on every aspect, the main difference between them being that the former is better in terms of safety and the latter offers the best performance in travel time.

5.7.3 The Zaragoza Tramway

The use of more than one model in a traffic engineering study is becoming increasingly commonplace. A typical combination is the use of macroscopic models to determine or/and refine travel demand (origin–destination matrices) with micro (or meso)-simulation taking this data as an input for disaggregate assessments that focus on operational problems. The case study presented in this section highlights some interesting limitations in the typical 'top-down' implementation of macro/micromodelling. We discuss briefly an iterative approach which we used successfully to evaluate the future impact of a proposed tramway in the city of Zaragoza, Spain.

The root of the issue we will discuss lies in the fact that macro and dynamic models are usually not applied at the same scale level. Macromodels are typically applied

to an entire city, while micro-simulation is most often used to analyse a sub-part of the same road network. To do so, OD demand for the micro area is determined using what is called a 'traversal' generation (reference). This consists in assigning the traffic to the general network with the static model; capturing the trips that enter, exit or stay within the subarea and finally generating the sub-matrix. The underlying risk of such an approach is that, during the microscopic simulation phase, a user could decide to evaluate changes that affect the validity of the demand matrix. Consider, for example, a new infrastructure design where an additional lane is added to a road section or the capacity of a given street is reduced by dramatically cutting the green time of a traffic signal or by implementing a bus lane. In such situations, working with the same sub-matrix for the base case and the future scenario is fundamentally incorrect. This is because if a change occurs that affects the supply conditions, it is highly probable that the traffic flows that used this part of the network will change as well. This necessitates that the demand of the subarea be computed again, moving away from the 'top-down' approach and adopting a macro–micro iterative one instead.

Returning to the specific case of Zaragoza in Spain, our study aim was to determine the impact of a new tramway in the city. The first step was to build a macromodel for the entire city with the purpose of determining the global OD matrix both for public transport and private cars. After calibrating and validating field measurements, we computed a traversal matrix to determine the demand in the subarea of the tramway corridor. Using the traversal demand as input, we then calibrated a micro-simulation model. The next step was to build a model of a future scenario that included the tramway line and the full-priority pre-emption system at intersections. The priority to the tram is a typical operational aspect for which micro-simulation is well suited as it allows a very accurate description of tram arrival detection and traffic signal setting changes. Once the model was ready, we micro-simulated, in the first instance using the traversal matrix of the base case. Results, illustrated in Fig. 5.23 (*left pane*), show important congestion in the centre of the area. This congestion is mostly due to the dramatic decrease of green time for the streets cutting the tramway line perpendicularly. This projection is plausible if these radical changes were to be implemented instantaneously with no notice. Taken at face value these results suggest that the new tramway will generate a critical decrease in performance for private vehicle flows. If used without further qualification, the outputs could lead to a rejection of this new public transport infrastructure initiative.

Applying the macro–micro iterative process, the street capacities that we computed in the micro-simulation of the future scenario were used as input to the macromodel and a new traversal matrix was calculated. We used the new sub-matrix to micro-simulate again the tramway corridor, providing results that can be shown in Fig. 5.23 (*right pane*). This iterative process, described in Fig. 5.24, allowed the traffic engineer in charge of this project to observe how the demand in the subarea has been adequately redistributed within the network. This re-distribution could be considered the long-term reaction of road users to the new traffic conditions. The revised micro-simulation results show now that the congestion identified in the original micro-simulation would mostly disappear. The revised results lead to a totally

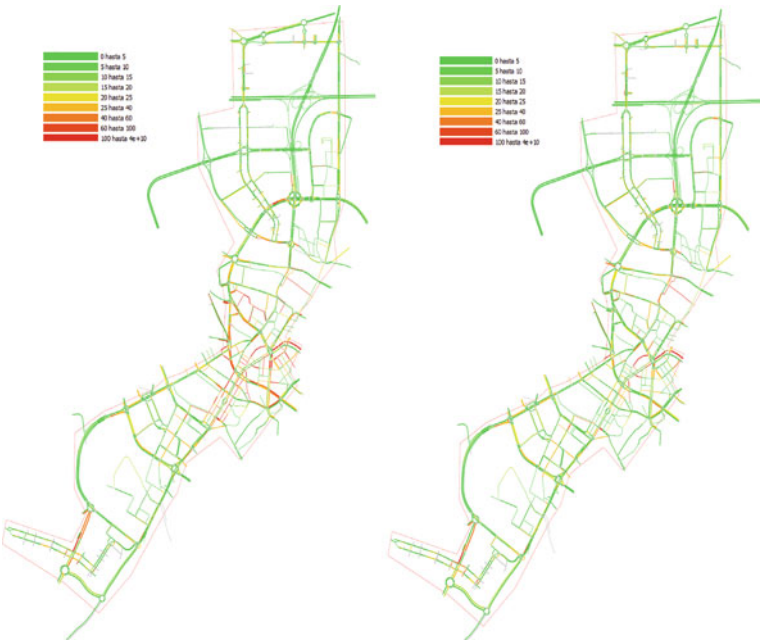


Fig. 5.23 Traffic density varying in the micro-simulated tramway corridor network before (short term, *left pane*) and after (long term, *right lane*) recalculating the local OD matrices and paths

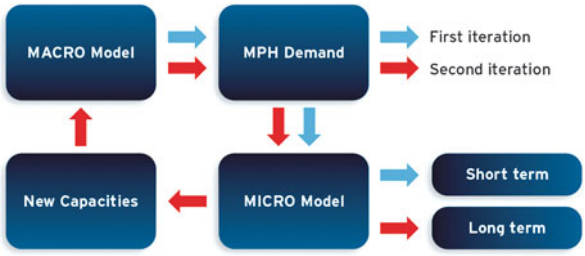


Fig. 5.24 Outline of the proposed iterative process

different conclusion whereby the new tramway has a limited long-term impact on car traffic making its implementation quite viable.

This Zaragoza case study illustrates the merit of adapting one's approach to the problem's parameters rather than following a top-down approach.

5.8 Modelling Details of Advanced Case Studies

When moving to real-time traffic management decision support, complexity increases making this an ideal field for the application of rich methodologies that apply several models sequentially, iteratively or even concurrently. To illustrate

some of the possibilities, this section describes the concept of the Aimsun Online solution and discusses its implementation in Madrid as an example. In the latter part of the section, we comment on some challenges related to such applications and the development needs that they give rise to.

5.8.1 The Aimsun Online Application in Madrid

The proliferation of ITS applications makes real-time prediction capabilities a concrete requirement for the dynamic management of networks, in both urban and interurban environments. Numerous techniques for real-time forecasting have been developed and some of them have been implemented around the world. However, considering the complexity and dynamicity of problems faced by traffic managers, aggregate solutions relying on time series analysis of detector data or static equilibrium assignment techniques tend to have very limited applicability and benefit. On the other hand, traffic simulation is increasingly able to deal with very large networks, former computational and data availability limitations no longer providing a barrier to its application in real time.

In dynamic traffic management cases, the forecasting capabilities can be used to either disseminate information to travellers or, more usefully, compare the performance of different management strategies in response to a congestion and take the most appropriate action. Simulation-based systems are intrinsically better than aggregate solutions in dealing with non-recurrent events because fluctuations in supply can be explicitly factored in, and their impact under different scenarios can be quantified. These scenarios are composed of a set of actions – examples would be a lane closure, rerouting with VMS or speed limit variation – that can be activated manually or automatically based on rules which constantly process detector data. For recurring or predictable incidents, management scenarios are already implemented in the simulation model (within a scenario catalogue) and can be activated rapidly when the performance of a particular scenario has to be assessed in real time through simulation. The measures of effectiveness (MOEs) used to compare response strategies vary by project and may encapsulate safety, environmental, economic and operational considerations. A multi-objective scenario comparison, for example, may point out the scenario offering the lowest global travel time out of the ones which avoid any vehicles queuing at a specific and safety-sensitive tunnel.

A real-time simulation-based decision support tool based on Aimsun and called ‘Aimsun Online’ has been implemented in the Madrid traffic control centre. The newly opened M-30 urban highway (composed of a significant number of tunnel sections) is subject to many safety considerations, and many traffic evacuation and rerouting actions may be applied in response to incidents. For this reason, a tool capable of anticipating the consequences of these actions on the neighbouring network over the following critical 15 or 30 min was necessary. The tool allows operators to choose which set of actions on the city can support these safety measures efficiently while minimizing the impact on the rest of the traffic.

This application is fed field measurements in real time and uses this data to deduce the current traffic status on the streets and the actual demand (modelled as an origin–destination matrix). With signal control plans changing dynamically during the day, the application also reads the current control plan operated at each network intersection. M-30 safety actuations as well as any other incident previously detected and still existing are automatically (and in some case manually) loaded into the simulation model before starting the parallel simulation runs. Each simulation considers a concrete set of actions (strategy) that might be applied in order to improve the network situation compared to the 'do-nothing' case.

Once the parallel simulations have completed, which vary between 1 and 3 min after the initial call, operators are supplied with a summarized view of the results; this includes snapshots of predicted traffic congestion and performance indicators (MOEs). These results ultimately allow the operator to quickly see, first, if any strategies improve the situation compared to the 'do-nothing' case and if yes, which ones offer the best performance. If the suggested solution is accepted, a single validation click on the screen leads to the field application of the selected strategy. An offline-operated module allows the prediction capabilities of the simulation to be evaluated each day by comparing simulation results against real data stored during the day, offering the City Council a measure of confidence in the reliability of the system (Fig. 5.25).

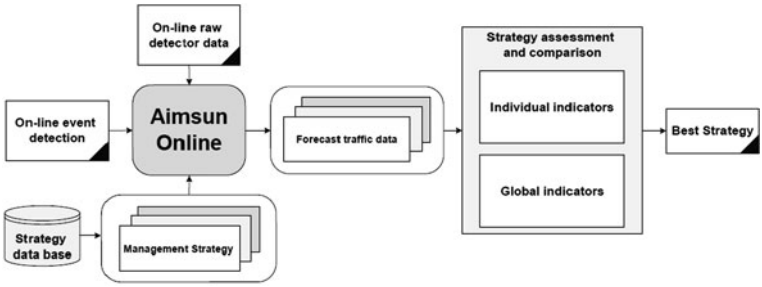


Fig. 5.25 Schematic view of the use of simulation for real-time traffic management decision support

5.8.2 Challenges and Further Needs

The Aimsun Online solution addresses a set of traffic management problems for which a combination of macro and dynamic models (micro and/or meso) is extremely well suited. The role of the macromodel here is mainly limited to OD matrix operations and more specifically, adjustments based on field measurements. However, these techniques are not without limitations and those should be borne in mind when applying them in a real-time context. To take one example, it is important to keep in mind that traffic volumes detected in the field do not represent

true demand levels under high congestion; in such cases, detector data reflect only supply-side information – but not demand, which is higher. This information should therefore be treated with special attention. Using dynamic models (especially the mesoscopic model) for matrix adjustment could offer, among other advantages, an interesting solution to this problem as the demand–supply relationship would be realistically represented. Implementation of dynamic matrix adjustment solutions is therefore, in the authors' view, a clearly identified need for the future.

Another delicate aspect of such simulation-based decision support solutions is the dynamic traffic assignment. As commented in an earlier section, although the concept of dynamic user equilibrium (DUE) is well suited to describing recurring, steady-state traffic patterns, its ability to correctly represent the short-term impact on traffic distribution of a non-recurrent incident is debatable. In this case, it is indeed fair to consider the flows no longer in equilibrium and that a route choice model being able to represent the behaviour of drivers under provision of partial or complete traffic information would be more appropriate. Based on this reasoning, a careful combination of DUE-based paths and the stochastic route choice in the same run of a dynamic simulation is an ideal solution with vehicles moving from one type of assignment to the other as a function of time and information available to them.

Finally, the use of Aimsun Online always gives rise to the same debate: Which dynamic modelling approach should be used? Micro or meso? It is hardly controversial to state that each model has its own advantages and disadvantages. The microscopic simulator is able to represent section and node dynamics in detail making it suitable for ITS applications in which such granularity of information is not just useful but, one would argue, essential. In addition, a microscopic model offers a larger variety of disaggregate outputs (environmental ones, for example). However, it has the limitation of important calibration effort needs and slower computational performance. The mesoscopic approach, on the other side, is the ideal tool for fast simulations which are definitively needed in real-time applications. However, adaptive signal control and bus priority systems together with bus stops and pedestrian crossings are examples of aspects that are only approximately, if at all, modelled mesoscopically. Therefore, there is no definitive answer to this question. Depending on the network characteristics, the objectives and the level of expected accuracy micro or meso should be chosen as a compromise. However, the simultaneous use of both micro- and mesomodels coupled with appropriate combinations of dynamic traffic assignment schemes would enjoy the benefits of both categories without their limitations. In that sense, a hybrid meso–micro represents a 'best of both worlds' and as such constitutes a major need for future developments in this field and a key tenet of Aimsun's development path.

Acknowledgment The authors and the rest of the staff at TSS – Transport Simulation Systems – would like to express their sincere gratitude to Professor Jaume Barceló at the Technical University of Catalonia (UPC) for his numerous and varied contributions to the inception, design and evolution of Aimsun over the years. This chapter would exist without him.

References

Aimsun Version 6 User's Manual (2008) TSS—transport simulation systems, Barcelona, Spain, www.aimsun.com

Alexiadis V (2007) Role of simulation in corridor management. Presented at TRB2007 simulation workshop, Washington, DC, January

Balci O (1998) Verification, validation and testing. In: Banks J (ed) Handbook of simulation: principles, methodology, advances, applications and practice. John Wiley & Sons-Interscience, New York, US

Barceló J, Casas J (2002) Dynamic network simulation with AIMSUN. Presented at the international symposium on transport simulation, Yokohama (also in: Kitamura R, Kuwahara M (eds) Simulation approaches in transportation analysis: recent advances and challenges. Kluwer, 2005)

Barceló J, Casas J (2004a) Heuristic dynamic assignment based on AIMSUN microscopic traffic simulator. Proceedings of TRISTAN V, Guadeloupe

Barceló J, Casas J (2004b) Methodological notes on the calibration and validation of microscopic traffic simulation models. Transportation Research Board, 83rd annual meeting, Washington, DC, 2004

Barceló J, Casas J (2006) Stochastic heuristic dynamic assignment based on Aimsun microscopic traffic simulator. Transport Res Rec 1964:70–79

Barceló, J, Ferrer JL, Grau R (1994) AIMSUN2 and the GETRAM simulation environment. Research report, Departamento de Estadística e Investigación Operativa. Facultad de Informática, Universidad Politécnica de Cataluña

Barceló J, Casas J, Ferrer JL, García D (1998a) Modeling advanced transport telematic applications with microscopic simulators: the case of AIMSUN2, simulation technology, science and art. In: Bargiela A, Kerckhoffs E (eds.) Proceedings of the 10th European simulation symposium. Society for Modeling and Computer Simulation International, Vista, California, US, pp 362–367

Barceló J, Ferrer JL, García D, Florian M, Le Saux E (1998b), Parallelization of microscopic traffic simulation for ATT systems analysis. In: Marcotte P. Nguyen S (eds) Equilibrium and advanced transportation modeling. Kluver Acadmic Publishing, Boston/Dordrecht/London

Barceló J, Casas J, García D, Perarnau J (2006) A hybrid simulation framework for advanced transportation analysis. International symposium on traffic simulation, ISTS 2006, Lausanne, Switzerland

Ben-Akiva, M, Bierlaire M (1999) Discrete choice methods and their applications to short term travel decisions. In: Hall RW (ed) Handbook of Tranportation science. Kluver Academic Publishers, Boston/Dordrecht/London

Bleile T, Krautter W, Manstetten D, Schwab T (1996) Traffic simulation at Robert Bosch GmbH. Proceedings of the Euromotor seminar telematic / vehicle and environment, Aachen, Germany, Nov. 11–12

Carey M, Ge YE (2007) Comparison of methods for path flow reassignment for dynamic user equilibrium, May 2007. School of Management & Economics, Queen's University, Belfast, Northern Ireland

Cascetta, E, Nuzzolo A, Russo F, Vitetta A (1996) A modified logit route choice model overcoming path overlapping problems. In: Proceedings of the 13th international symposium on transportation and traffic flow theory. Pergamon Press, Oxford, UK

Ferrer JL, Barceló J (1993) AIMSUN2: advanced interactive microscopic simulator for urban and non-urban networks. Research report. Departamento de Estadística e Investigación Operativa, Facultad de Informática, Universidad Politécnica de Cataluña

Florian M, Hearn D (1995) Network equilibrium models and algorithms. In: Ball MO et al. (eds) Handbooks in OR and MS, Chapter 6, vol.8. Elsevier, Amsterdam, Netherlands

Florian M, Mahut M, Tremblay N (2001) A hybrid optimization–mesoscopic simulation dynamic traffic assignment model. Proceedings of the 2001 IEEE intelligent transport systems conference, Oakland, pp 118–123

Florian M, Mahut M, Tremblay N (2002) Application of a simulation-based dynamic traffic assignment model. Presented at the international symposium on transport simulation, Yokohama (also in: Kitamura R, Kuwahara M (eds) Simulation approaches in transportation analysis. Kluwer, 2005, pp 1–21)

Friesz TL, Bernstein D, Smith TE, Tobin RL, Wie BW (1993) A variational inequality formulation of the dynamic network user equilibrium problem. Oper Res 41(1):179–191

Gipps PG (1981) A behavioural car-following model for computer simulation. Transport Res Board 15-B:105–111

Gipps PG (1986a) A model for the structure of lane-changing decisions. Transport Res Board 20-B(5):403–414

Gipps PG (1986b) MULTSIM: a model for simulating vehicular traffic on multi-lane arterial roads. Math Comput Simul 28:291–295

Janson BN (1991) Dynamic assignment for urban road networks, Transport Res B 25(2/3):143–161

Kleijnen JPC (1995) Theory and Methodology: Verification and Validation of Simulation Models, European Journal of Operational Research, vol. 82, pp. 145–162. Elsevier, Amsterdam, Netherlands

Law AM, Kelton WD (1991) Simulation modeling and analysis, 2nd edn. McGraw-Hill, New York, USA

Liu HX, Ma W, Ban JX, Michardani P (2005) Dynamic equilibrium assignment with microscopic traffic simulation. 8th international IEEE conference on intelligent transport systems, Vienna, Austria

Lo HK, Szeto WY (2002) A cell-based variational inequality formulation of the dynamic user optimal assignment problem. Transport Res B 36:421–443

Mahmassani H (2001) Dynamic network traffic assignment and simulation methodology for advanced system management applications. Netw Spatial Econ 1:267–292

Mahut M (1999a) Speed-maximizing car-following models based on safe stopping rules. Transportation Research Board, 78th annual meeting, January 10–14, 1999 Washington DC, US

Mahut M (1999b) Behavioural car following models. Report CRT-99-31. Centre for Research on Transportation, University of Montreal, Montreal, Canada

Mahut M (2001) A discrete flow model for dynamic network loading. Ph.D. thesis, Department d′IRO and CRI, University of Montreal

Manstetten D, Krautter W, Schwab T (1998) Traffic simulation supporting urban control system development. Proceedings of the 4th world conference on ITS, Seoul

Peeta S, Mahmassani H (1995) System optimal and user equilibrium time-dependent traffic assignment in congested networks. Ann Oper Res 60:81–113

Ran B, Boyce D (1996) Modeling dynamic transportation networks. Springer, Berlin, Germany

Rouphail NM, Sacks J (2003) Thoughts on traffic models calibration and validation. Paper presented at the workshop on traffic modeling, Sitges, Spain

Sbayti H, Lu C, Mahmassani H (2007) Efficient implementations of the method of successive averages in simulation-based DTA models for large-scale network applications. TRB 2007 annual meeting, Chicago, Illinois, US

Sheffi Y (1985) Urban transportation networks. Equilibrium analysis with mathematical programming methods. Prentice-Hall, Englewood Cliffs, NY

Smith MJ (1993) A new dynamic traffic model and the existence and calculation of dynamic user equilibria on congested capacity-constrained road networks. Transport Res Part B 27:49–63

Stogios Y, Pringle R, Nikolic G (2008) A traffic simulation framework for the greater Toronto area freeway network: concept and challenges. TRB 2008 annual meeting, Washington DC, US

Theil H (1966) Applied economic forecasting. North-Holland, Amsterdam, Netherlands

Tong CO, Wong SC (2000) A predictive dynamic traffic assignment model in congested capacity-constrained road networks. Transport Res B 34:625–644

Torda A, Casas J, Garcia D, Perarnau J (2009) Traffic Simulation as a Central Element for Traffic Management Decision Support Systems. Second International Symposium on Freeway and Tollway Operations (ISFO), Hawai, USA

Varia HR, Dhingra SL (2004) Dynamic user equilibrium traffic assignment on congested multidestination network. J Transport Eng 130(2):211–221

Yoshii T (1999) Standard verification process for traffic simulation model – verification manual. Kochi University of Technology, Kochi, Japan

Chapter 6
Traffic Simulation with MITSIMLab

Moshe Ben-Akiva, Haris N. Koutsopoulos, Tomer Toledo, Qi Yang, Charisma F. Choudhury, Constantinos Antoniou, and Ramachandran Balakrishna

6.1 Introduction

MITSIMLab (microscopic traffic simulation laboratory) is a microscopic traffic simulation model that evaluates the impacts of alternative traffic management system designs, traveler information systems, public transport operations, and various ITS strategies at the operational level and assists in their subsequent refinement. MITSIMLab can evaluate systems such as advanced traffic management systems (ATMS) and route guidance systems.

MITSIMLab was developed by MIT's Intelligent Transportation Systems (ITS) Program (Yang, 1997; Yang and Koutsopoulos, 1996; Yang et al., 2000). The model

M. Ben-Akiva (✉)
Massachusetts Institute of Technology, 77 Massachusetts Ave., Rm. 1-181, Cambridge, MA 02139, USA
e-mail: mba@mit.edu

H.N. Koutsopoulos
The Royal Institute of Technology, KTH Teknikringen 72, SE - 100 44 Stockholm, Sweden
e-mail: hnk@infra.kth.se

T. Toledo
Technion – Israel Institute of Technology, Haifa 3200, Israel
e-mail: toledo@technion.ac.il

Q. Yang
Caliper Corporation, 1172 Beacon Street, Suite 300, Newton, MA 02461, USA
e-mail: qiyang@caliper.com

C.F. Choudhury
Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh
e-mail: cfc@ce.buet.ac.bd

C. Antoniou
National Technical University of Athens, 9 Heroon Politechniou St., GR-15780 Athens, Greece
e-mail: antoniou@central.ntua.gr

R. Balakrishna
Caliper Corporation, 1172 Beacon Street, Suite 300, Newton, MA 02461, USA
e-mail: rama@caliper.com

J. Barceló (ed.), *Fundamentals of Traffic Simulation*,
International Series in Operations Research & Management Science 145,
DOI 10.1007/978-1-4419-6142-6_6,

was used to evaluate several aspects of the traffic management system for the Central Artery/Tunnel (CA/T), by simulating its operations. The CA/T network consists of approximately 110 lane miles equipped with 1600 sensors and is used by 300,000 vehicles per day. It features an extensive traffic control system, including lane control signals (LCS), incident detection, tunnel closing, electronic toll collection (ETC), and variable message signs (VMS) for route guidance. MITSIMLab was used to evaluate various operating strategies associated with these traffic management functions and make recommendations for improvements, including ITS design, ramp configuration, and construction staging. For the CA/T application, MITSIMLab was calibrated with behavior data of Boston drivers.

MITSIMLab serves as a laboratory for the evaluation of ITS and other traffic and transit strategies and systems. The model's application framework for these evaluations is outlined in Fig. 6.1. Based on the objectives of the evaluated system, scenarios are generated to test the design. Appropriate measures of performance are generated from the simulations, used to evaluate the system performance, and may lead to subsequent design refinements.

MITSIMLab supports the following:

- Objective and independent evaluations.
- Thorough representations of all relevant interactions in the transportation system, including vehicles, traffic control devices, algorithms, and other elements of the traffic management center (e.g., surveillance system).
- Assessment of the technical aspects of the algorithms, the performance and impact of interfaces and communication channels, sensitivity to errors, robustness, and ability to recover from malfunctions.

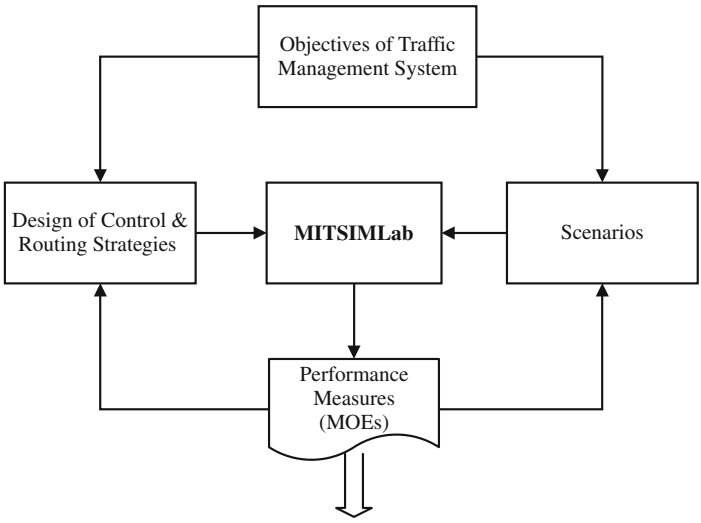


Fig. 6.1 Evaluation framework

MITSIMLab represents the related functions of the traffic management system at a fine level of detail, including the important aspects of the traffic management center, the surveillance system, the guidance and control logic, and algorithms, in order to evaluate a wide range of design aspects of ATIS/ATMS. Researchers have used MITSIMLab for practical applications in the USA, the UK, Sweden, Italy, Switzerland, Japan, Korea, Malaysia, and elsewhere. It was the main tool used to test and demonstrate the various driving behavior models developed within the NGSIM (next generation simulation) project, which facilitated the advancement of traffic simulation models by improving realism in the driving behavior models they incorporate. In 2004, an open-source version of MITSIMLab was released. It is available at the MIT ITS Program website (http://mit.edu/its/mitsimlab.html). The structure and models in MITSIMLab also formed the basis for the development of the traffic simulation software TransModeler (www.caliper.com/transmodeler/default.htm).

This chapter describes the structure and main characteristics of MITSIMLab, the methodology used for model calibration and validation, followed by several application examples.

6.2 Model-Building Principles in MITSIMLab

In order to allow maximum flexibility in defining the evaluated systems, travelers' behavior in the presence of these systems, and the dynamic interactions between the management system and travelers, MITSIMLab is implemented as three separate modules, which exchange information as shown in Fig. 6.2.

Within the traffic simulator (TS) module, the movements of individual vehicles (cars and transit vehicles) are represented by detailed travel and driving behavior models. Traffic flow characteristics emerge from these individual behaviors. Vehicles traveling in the network activate surveillance devices (e.g., loop detectors,

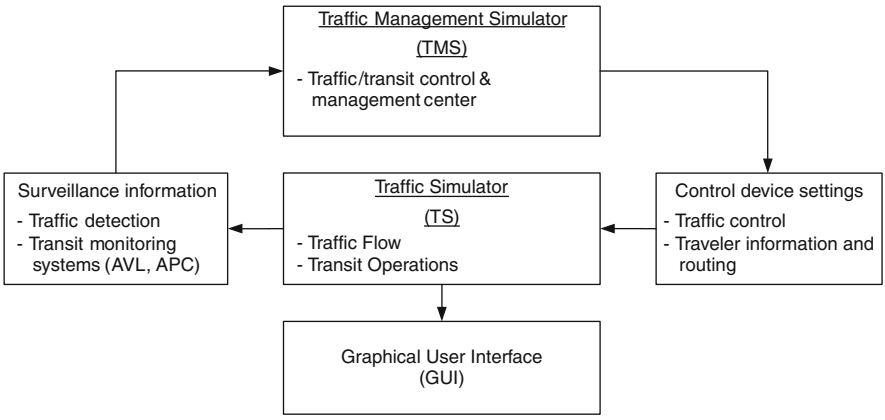


Fig. 6.2 MITSIMLab structure

communication beacons, video sensors). The data gathered by the surveillance system are transferred to the traffic management simulator (TMS), which mimics the traffic control and routing strategy or transit strategy under evaluation. The control and routing strategies generated by the TMS determine the states of traffic control and route guidance devices. These settings are transferred to the TS. The simulated drivers respond to the various traffic controls and guidance, while interacting with each other. The TMS is a virtual transportation system operation control center, processing performance data from the sensor network and generating a strategy. The TMS also simulates a wide range of transit operations control strategies (e.g., transit signal priority and holding for service restoration) defined by the user. The simulation output can be obtained as numerical data tables and via the graphical user interface (GUI), which visualizes traffic impacts through vehicle animation. MITSIMLab generates various output reports with measures of effectiveness that may be used to evaluate the performance of potential ITS strategies.

Travel demand is represented by time-dependent origin-to-destination (OD) trip tables, which show expected conditions or are defined as part of a scenario for evaluation. Based on these tables, individual vehicles are generated. The generated vehicles are assigned driver characteristics (e.g., aggressiveness, planning capability, look-ahead distance, level of compliance with various signs and regulations) and vehicle attributes (e.g., acceleration and speed capabilities and the impact of grade on these capabilities) based on pre-determined distributions. Route choices are based on a probabilistic model that captures the impact of travel times and biases toward routes that use freeways over urban streets. The impact of real-time information on routing decisions is captured by a route-switching model in which informed drivers re-evaluate their pre-trip route choices based on the traffic conditions observed en route. MITSIMLab is a time-based simulation model with time steps that may differ for various functions from 0.1 to 1.0 s. It also incorporates event-based approaches for situations such as crash avoidance and responses to changes in traffic controls and information settings.

6.3 Fundamental Core Models

The core of MITSIMLab consists of travel and driving behavior models. The travel behavior models capture the driver's pre-trip and en route route choices. The driving behavior models deal with tactical and operational driving decisions, mainly acceleration and lane changing. The models that capture these choices in MITSIMLab are probabilistic, based on the theories of random utility maximization.

6.3.1 Driving Behavior

Driving behavior decisions are modeled as a series of interdependent choices that are based on a specific plan/tactic. For example, drivers select a target lane and

adapt their acceleration and lane-changing actions to facilitate arriving at the chosen lane. The evolving circumstances (i.e., behavior of other drivers, traffic control) can cause changes to the plan. For example, drivers may initially plan to merge into mainline traffic through normal gap acceptance. But as they approach the end of the merging lane and are unable to find acceptable gaps, they may force merge. Drivers' plans are generally unobservable in the real world (only drivers' actions are observed). Therefore, MITSIMLab captures this behavior using an integrated modeling framework based on latent plans.

The general framework of these models is shown in Fig. 6.3. At any instant, drivers choose a plan based on the state they are faced with. Their actions depend on the chosen plan. These actions, the actions of the other drivers, and changes in the state of the control system (e.g., traffic light indications) may lead drivers to change their plans.

The plan drivers' choice may depend on their previous plan choices and be affected by anticipated future conditions. The models that capture the plan choice and the action choice, conditional on this plan, are based on the utility maximization theory. The interdependencies and causal relationships between various decisions over time and across choice dimensions result in serial correlation and state dependence among the observations. Driver-specific random terms are incorporated into the models in order to capture heterogeneity in drivers' behavior that stems from differences in aggressiveness, planning capabilities, etc. A hidden Markov model is used to capture the effect of previously chosen plans on the choice of the current one. Effects of anticipated future circumstances are captured using predicted conditions based on current information in the decision making. For a complete description of the latent plan model structure, see Choudhury (2007).

The main driving behavior models in the latent plan structure are lane changing and acceleration. In MITSIMLab, they are modeled using an integrated framework

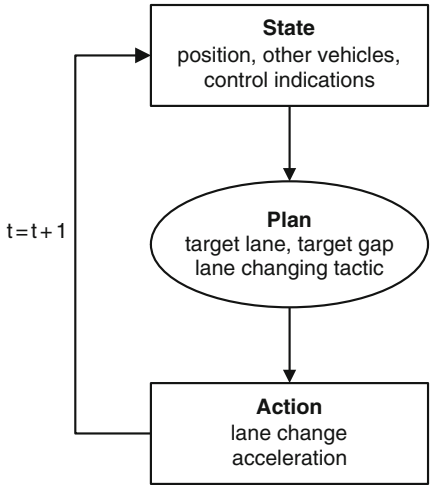


Fig. 6.3 General decision structure

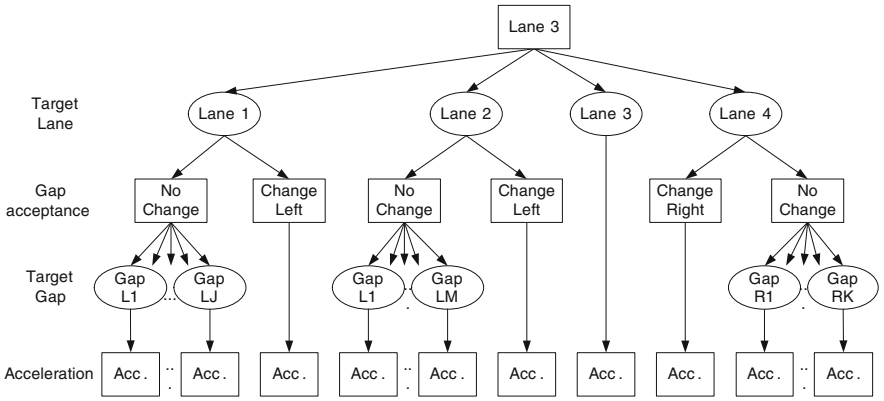


Fig. 6.4 Structure of the driving behavior model

as shown in Fig. 6.4. The figure shows the decision process for a driver currently in lane 3 of a four-lane road. Chosen (latent) plans are shown as ovals and the resulting actions as rectangles.

Drivers select the target lane as the lane they perceive to be the best among all available lanes (lanes 1, 2, 3, and 4 in this case).

The target lane model uses a multinomial logit (MNL) structure. Important variables that affect lane choices include the distance to the point where the driver must be in specific lanes in order to follow the path, the number of lane changes required to be in these lanes, the attributes of the various lanes (e.g., average speed and density of the lane, lane cost/toll), and variables that capture the conditions in the immediate vicinity of the vehicle such as the relative speeds and spacing from lead vehicles, the presence of heavy vehicles, and characteristics of the driver (e.g., aggressiveness, network familiarity). If the target lane is different from the current lane, a lane change is required. Drivers then search for an acceptable gap to complete the lane change.

The gap acceptance model in MITSIMLab is probabilistic, where the available gaps are compared against the critical gaps. The model defines the lead and lag gaps as the clear spacing between the subject vehicle and the lead and lag vehicles in the adjacent lane, respectively. A gap is acceptable only if the lead and lag gaps are acceptable (i.e., available gap = critical gap). Critical gaps are assumed to be log-normally distributed, where the mean is a function of explanatory variables, which include the relative speeds of the lead and lag vehicles.

The drivers that cannot change lanes immediately select a short-term plan to perform the desired lane change. Short-term plans are defined by the various traffic gaps in the target lane. The target gap choice probabilities are modeled with an MNL structure where the trade-offs among different attributes of the gap (e.g., gap size and distance to the gap) are accounted for.

Drivers adapt their acceleration behavior to facilitate their short-term plans (i.e., target lane and gap). Different accelerations are applied depending on the current

plan the driver implements: stay in the lane, lane changing, or target various gaps for lane changing. The stimulus-sensitivity framework proposed within the GM model (Gazis et al., 1961) is adapted for these acceleration models. The response (acceleration or deceleration) the driver applies to a stimulus is lagged to account for reaction time as follows:

$$\text{response}_n(t) = \text{sensitivity}_n(t) \times \text{stimulus}_n(t - \tau_n) \tag{6.1}$$

where t is the time of observation and τ_n is the reaction time for driver n.

The driver reacts to different stimuli depending on the chosen plan and constraints imposed. Within each one of the acceleration behaviors, the driver is assumed to be in either a constrained or an unconstrained regime. A constrained regime applies when the driver is close to the lead vehicle (the headway is smaller than the threshold) and affected by its behavior. For stay-in-lane and target gap accelerations, the lead vehicle is the front vehicle in the current lane. For lane-changing acceleration, the lead vehicle is the front vehicle in the lane the driver is changing to. In the constrained regime, the stimulus is the relative speed of the lead vehicle and has different parameters for acceleration and deceleration. In the unconstrained regime, for the stay-in-the-lane and lane-changing cases, free-flow acceleration is applied. For target gap cases, the stimulus is determined by a desired position that would facilitate completion of the lane change. The reaction time and time headway threshold distributions account for the heterogeneity among drivers and are common to all components of the acceleration model. See Ahmed (1999) and Toledo (2008) for details of the target gap choice and acceleration models.

One of the main factors affecting lane choices is the need to follow the travel path. The implementation of path awareness (i.e., when do drivers become aware and begin to respond to path-following constraints) impacts the simulation results. The path awareness model in MITSIMLab assumes that drivers are aware of the path plan up to a certain distance downstream of their current position. They will react to any path-following constraints that arise within this "look-ahead" distance and ignore those that are further downstream. The look-ahead distances are characteristics of the driver and are assumed to be randomly distributed in the driver population. This approach overcomes the excess weaving and merging maneuvers arising from late lane changes that occur when the awareness is based on the network structure (i.e., drivers are only aware of the next link(s) on their path), particularly in urban networks that are characterized by short links and paths that may require frequent turning movements.

6.3.2 Travel Behavior

The travel behavior models include both pre-trip and en route path choices. Drivers in MITSIMLab may either have predefined paths or compute them dynamically. Depending on whether alternative paths were predefined, a path-based route choice model or a link-based route generation model may be used.

In the path-based route choice model, a list of predefined paths is used as input. Each path is defined by the list of links it consists of. The choice among these lanes is modeled with the path-size model (Ben-Akiva and Bierlaire, 2003), which accounts for the similarity among paths that overlap in parts. With this model, the probability that a driver will choose route i from the path choice set C is given as

$$P(i) = \frac{\exp\left(V_i + \ln \mathrm{PS}_i\right)}{\sum_{j \in C} \exp\left(V_i + \ln \mathrm{PS}_i\right)} \tag{6.2}$$

where, V_i and V_j are the systematic utilities of routes i and j, respectively. The systematic utilities in the route choice model are functions of path attributes such as path travel times and freeway bias. Travel times may be habitual or predicted. PS_i and PS_j are the corresponding path sizes. These terms capture the effect of overlapping routes on drivers' perceptions.

The link-based route generation model does not require path enumeration, which may be expensive in large urban networks. Instead, it represents a myopic behavior. This model is also useful in generating an initial set of paths between origins and destinations. With this model, drivers choose only the next link at each intersection. An MNL model is used for this choice:

$$P\left(k|s,d\right) = \frac{\exp\left(V_{kd}\right)}{\sum_{j \in L_s} \exp\left(V_{jd}\right)} \tag{6.3}$$

where, $P\left(k|s,d\right)$ is the probability of choosing link k as the next link on the path to destination d at node s. L_s is the set of links emanating from node s. V_{kd} is the systematic utility associated with link k for getting to destination d.

Since link travel times are time dependent, path travel times account for the time that drivers are expected to arrive at each link on their path. The travel time in the utility for each alternative link is the travel time from node s through the specific link to the destination, as illustrated in Fig. 6.5:

$$TT_{skd}\left(t\right) = tt_{sk}\left(t\right) + TT_{kd}\left(t + tt_{sk}\left(t\right)\right) \tag{6.4}$$

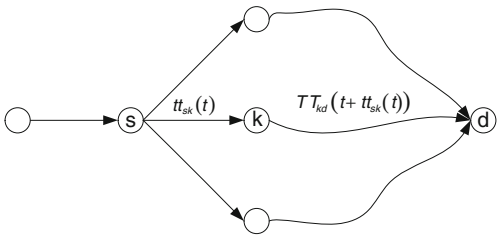


Fig. 6.5 Travel times in the link-based model

where $TT_{skd}(t)$ is the travel time to d using link k for vehicles arriving at s at time t. $tt_{sk}(t)$ is the travel time on the link sk for vehicles entering the link at time t. $TT_{kd}(t + tt_{sk}(t))$ is the travel time from k to the destination on the shortest path at the time the vehicle arrives to k.

To avoid using very long or circular paths, the link-based model uses additional parameters to screen out the choices leading to paths that are considered unrealistic. The first screening criterion removes alternatives with travel times that are too long compared to the alternative with the shortest travel time. A second screening criterion prevents vehicles from moving to nodes that are farther away from the destination compared to their current position.

The effects of traveler guidance and information on route choices are captured in the path-based and the link-based route choice models. Drivers in MITSIMLab are classified as informed or uninformed. In the presence of traffic information, informed drivers base route choices on updated travel times that incorporate real-time traffic conditions. If en route traveler information is available (e.g., through in-vehicle units or VMS), route choices are re-evaluated whenever new information is received. In the path-based model, drivers' preferences to keep their previously chosen paths are captured by a diversion dummy variable, which penalizes switching from the previously chosen route. Uninformed drivers use the habitual travel times, which represent prevailing traffic conditions on these paths. For networks with prescriptive VMS and route guidance systems, a compliance factor is defined to account for the fact that not all drivers adhere to the prescribed route.

6.3.3 Traffic Control

MITSIMLab, through the traffic management simulator (TMS), mimics the traffic control system in the evaluated network. A wide range of traffic control and route guidance systems can be simulated:

1. Ramp control
2. Freeway mainline control
   - Lane control signs (LCS)
   - Variable speed limit signs (VSLS)
   - Portal signals (PS) at tunnel entrances
3. Intersection control
4. Variable message signs (VMS)
5. In-vehicle route guidance

The TMS' general structure can represent different logical designs of such systems at varying levels of sophistication: from isolated pre-timed signals to real-time predictive systems. For example, a generic traffic controller is at the heart of the

TMS. The generic controller breaks down control strategies into basic logic elements and implements them within a modular framework. Specific control logic can then be recreated from these basic components. The modular structure allows any specialized features to be implemented easily. The logic behind the generic controller is specified in terms of signal groups (not phases). Each group is defined by the intersection movements that it controls and by the logic that governs its operation. A signal group holds data about its current status and its relationship to other groups, including its current indication (e.g., green arrow), its current action (e.g., holding the current period), the next indication to show (e.g., yellow arrow), its conflicting groups, and stored sensor data. In MITSIMLab, the status and position of every vehicle is updated at a specified step size (i.e., 0.1 s). A similar approach is used for the generic controller, which evaluates each signal group at every time step and determines if the state of any group needs updating. An overview of the logic of the generic controller is shown in Fig. 6.6.

Upon initialization, the controller obtains information about the signals that it will direct, the movements controlled by each signal group, the initial state of each signal group, and the conditions that specify the control logic for each signal group. During a simulation run, the controller iterates through all the signal groups, evaluating the logic conditions, and determines whether the group's state should be updated. This evaluation step is iterative because the group states may be interdependent with the state of one group being an input to the logic of another group. There are four types of conditions that correspond to different actions: *general* conditions that perform miscellaneous functions, *change* conditions that advance the signal group to the next period, *hold* conditions that keep the group in the current period, and *skip* conditions that indicate if the next specified number of conditions should be skipped. By combining the conditions in a specific order, a full controller logic can be specified. The types of control strategies that can be simulated include isolated controller operations (both fixed-time and demand-responsive) and coordinated operations (also both fixed-time and demand-responsive). A framework to incorporate fully adaptive control strategies has also been developed.

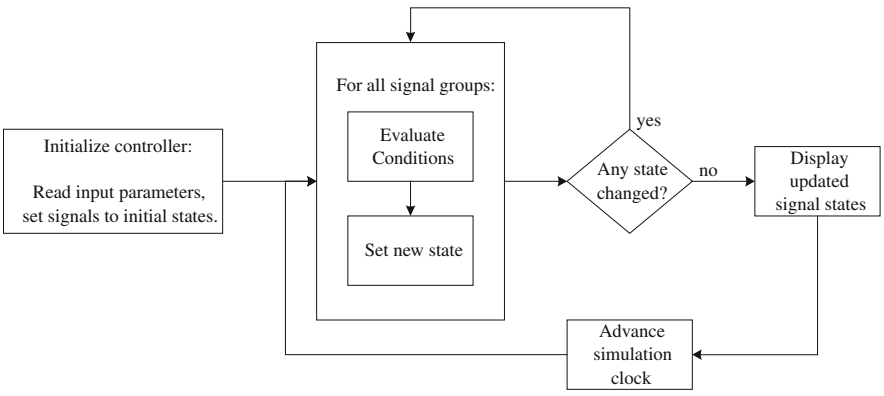


Fig. 6.6 Overall logic of generic controller

6.3.4 Transit Representation

The framework adopted to model bus operations benefits from MITSIMLab's modular organization. The main elements include the representation of the transit network, the movement of buses, the passenger demand, the transit surveillance system, and the operations of the transit control center.

The components of the transit system (transit network, schedule design, and fleet assignment) are considered as static information that is provided as input. A detailed representation of routes and schedules allows transit and traffic operations in the simulated network to be sensitive to the variations in the route and schedule inputs. MITSIMLab represents detailed trip chaining to explicitly capture the propagation of uncertainty in the network.

The driving behavior models control the transit vehicle movements. Specific transit operator models are applied in the sections between stops (considering downstream stops in the lane choice), when approaching stops (including undertaking lane changes to get to the stop lane), when departing from stops (merging to the general traffic lanes), and dwell times at stops. The behavioral models also incorporate the impact of the transit vehicle presence on the lane choices of other vehicles.

The representation of passenger demand determines the detail in passenger arrival and departure patterns on the transit network. A minimum representation of demand involves passenger impacts on dwell times at stops. This simplified representation ignores the impact of passenger interactions during boarding and alighting on bus progression, which affects dwell times downstream (since dwell times at a stop are independent of dwell times at stops upstream). The second level of demand representation uses arrival and alighting rates (defined as a percentage of the bus load) at stops to determine the numbers of boarding and alighting passengers. The model assumes that passengers arrive according to a probabilistic distribution (e.g., time-dependent Poisson) and randomly generates the number of passengers waiting to board based on the actual bus headway.

Transit surveillance and monitoring systems including onboard detection and sensing technologies, such as automated vehicle location (AVL) and automatic passenger counters (APC), are explicitly modeled.

Transit operations control center activities and decentralized field-deployed strategies are simulated in the TMS (see Fig. 6.2 above). The TMS mimics the logic of the strategy under evaluation and may use real-time traffic and transit data from the surveillance system as input for that logic. Device-based control strategies, such as signal priority, are simulated using sensors to detect approaching buses and to deliver bus data to the signal controller. Other strategies, such as stop-based control (e.g., holding), are simulated by placing conditions on the bus departure from a stop.

6.3.5 Measures of Performance

A number of measures of performance (MOPs) may be collected to characterize and evaluate the system. These measures may be defined at any level of detail for

the general and transit systems. Traffic-related outputs include flows, speeds, densities, travel time, delays, and queue lengths. They are available at the system, link, segment (a part of a link with uniform geometry), lane, sensor, and vehicle levels. The sensor-level data is particularly useful for calibration and validation of the simulation model against real-world data. The high level of detail of the collected individual vehicle data (positions at every 0.1 s of all vehicles) provides all the information necessary to develop statistics such as emissions and fuel consumption, which may be used for evaluation.

With respect to the transit system, a number of MOPs may be generated that are useful to assess the performance of the system both from a productivity point of view and from the passenger level of service perspective. As with general traffic, these MOPs may be in different levels of detail: system wide (e.g., total passenger travel times, number of late trips, driver overtime), route segments (e.g., average running speed, travel time distribution), stop (e.g., average dwell times), vehicle and passenger (e.g., waiting times and travel times).

6.4 Dynamic Traffic Assignment

MITSIMLab is not designed as a dynamic traffic assignment model and does not seek equilibrium travel time and traffic flow solutions automatically. It simulates drivers' route choice behavior based on input travel times and other path attributes. However, in the absence of habitual travel times, it requires an alternative method to assign vehicle trips to alternative paths from their origins to their destination. This method would consist of two related components: (1) determining the values of path attributes including the perceived link travel times and (2) computing the choice probabilities of the alternative paths. The models used to compute the path choice probabilities are described in Section 6.3.2.

A day-to-day learning process was used with MITSIMLab to estimate the congested link travel times. Multiple simulation runs were made. Each run represented a day. Travel times were updated as the weighted sum of the expected and experienced travel times from the current day (simulation run):

$$c_{it}^{(k+1)} = \lambda^{(k)}\, \hat{c}_{it}^{(k)} + \left(1 - \lambda^{(k)}\right) c_{it}^{(k)}, \qquad k = 0, 1, \ldots \tag{6.5}$$

where the indices i, t, and k are for the link, time interval, and simulation iteration (day), respectively. $\hat{c}_{it}^{(k)}$ and $c_{it}^{(k)}$ are the input (expected) and output (experienced) link travel times, respectively. $\lambda^{(k)}$ is a weighting parameter for iteration k, which may be determined, for example, according to the method of successive averages (Sheffi and Powell, 1982).

To support this functionality and to model response to real-time information, MITSIMLab maintains two time-variant travel time tables. The first represents the historical travel time associated with habitual route choices (input from another

model or study, or generated through the process outlined above). The second consists of the updated link travel times based on the real-time information system (if one is available).

6.5 Calibration and Validation

This section outlines the methodology that has been developed and applied for the calibration and validation of MITSIMLab, and the implemented behavioral models.

6.5.1 Overall Framework

Figure 6.7 illustrates the framework used for the model calibration and validation. The process uses both disaggregate and aggregate data. In the disaggregate calibration (or model estimation) phase, the behavioral model components (e.g., acceleration, lane changing, and route choice models) are estimated using detailed

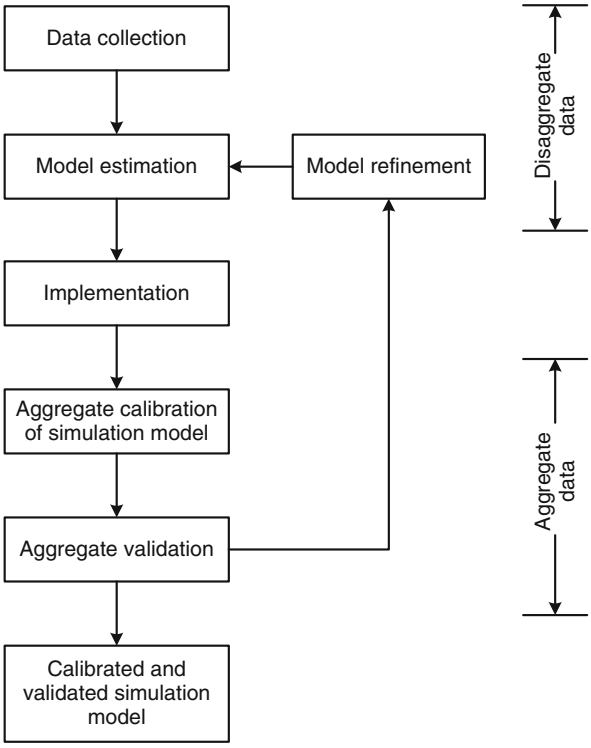


Fig. 6.7 Calibration and validation framework

data at the individual user level. For driving behavior models, the required data are vehicle trajectories at a high time resolution. For route choice models, the necessary data are the routes individual travelers have chosen. This estimation approach does not use traffic simulators, making the estimated models simulator independent.

In the aggregate calibration phase, the estimated models are calibrated jointly with other model components within MITSIMLab. This phase is also crucial if MITSIMLab is applied in a network where detailed trajectory data are not available, but aggregate data such as sensor counts and speeds are available. Part of the aggregate dataset is used to adjust key parameters in the behavior models and to estimate the travel demand on the case study network. This aggregate calibration problem is formulated as an optimization problem, which seeks to minimize a function of the deviation of the simulated traffic measurements from the observed measurements and of the deviation of calibrated values from their a priori estimates, if available (Toledo et al., 2004; Balakrishna et al., 2006b, 2007b). The rest of the data (to the extent possible, collected under different conditions) are used for the validation, which is based on comparisons of MOPs, calculated from the available data (e.g., sensor speeds and flows, the distribution of vehicles among the lanes, amount and locations of lane changes) with the corresponding values generated from the simulation runs.

6.5.2 Disaggregate Model Estimation

The disaggregate model estimation methodology is demonstrated through the estimation of a lane-changing model, which consists of drivers' lane selection and gap acceptance decisions (a simplified version of the model presented in Fig. 6.4). The specifications of the various components of this model are presented next, along with the resulting likelihood function to be maximized in the estimation.

The lane-changing maneuver is modeled as a two-stage process: (1) a choice of target lane (plan) and (2) a decision to accept available gaps (execution of plan). The target lane is the lane the driver perceives as the best while accounting for a wide range of factors and goals. A lane change is executed when the available lead and lag gaps are perceived as acceptable. An example of the structure of this lane-changing model is shown in Fig. 6.8. The decision maker is a driver currently in lane 3 of a four-lane road. The latent plan is captured by the choice of target lane. This latent choice dictates the immediate decisions of the driver; if the target lane is the same as the current lane (lane 3 in this case), no change is required. The direction of change is to the right if the target lane is lane 4, and to the left if the target lane is either lane 1 or lane 2. If the target lane choice dictates a lane change, the driver evaluates the gaps in the adjacent lane corresponding to the direction of change and either accepts the available gap and moves to the adjacent lane or rejects the available gap and stays in the current lane. In the trajectory data, the target lane choice is not observed. Only completed lane changes (or no changes) are observed. In Fig. 6.8, latent choices are shown as ovals and observed choices are represented as rectangles.

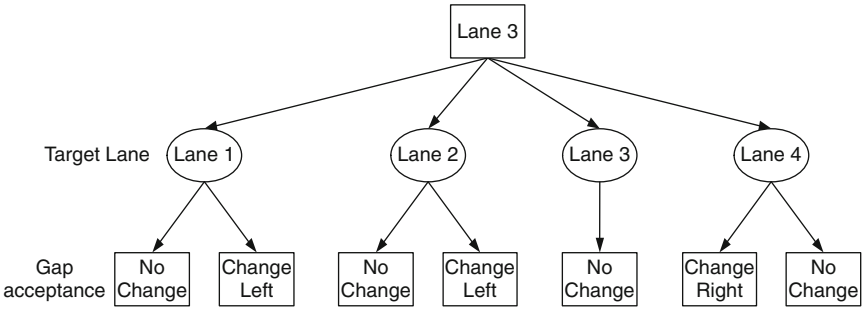


Fig. 6.8 Structure of the simplified lane-changing model

The latent plan choice is captured by the target lane. The target lane-choice set consists of all the available lanes the driver may travel in. The driver chooses the lane with the highest utility as the target. However, utilities are unobserved and modeled as random variables:

$$U_{int} = X_{int}\beta_i + \alpha_i \upsilon_n + \varepsilon_{int} \tag{6.6}$$

where U_{int} is the utility of lane i to individual n at time t. X_{int} and β_i are a vector of explanatory variables affecting the lane utility and the corresponding vector of parameters, respectively. υ_n and α_i are an individual-specific error term and the corresponding parameter, respectively. ε_{int} is a random error.

Different choice models are obtained depending on the assumptions for the distribution of ε_{int}:

$$P\left(TL_{int} \,|\upsilon_n\right) = g_i\left(X_{int}, \beta_i, \alpha_i, \upsilon_n\right) \tag{6.7}$$

where $g_i(.)$ is the function denoting the target lane choice.

The choice of target lane i dictates the change direction, d_i if one is required. If the current lane is also the target lane, no change is needed. Otherwise, the change will be in the direction of the target lane.

The gap acceptance model captures drivers' decisions in executing the chosen plan. That is, whether or not the available gap in an adjacent lane can be used to complete the desired lane change. To make this decision, the driver evaluates the available lead and lag gaps, which are defined by the free spacing between the subject and the lead and lag vehicles in the adjacent lane, respectively. The gap acceptance model assumes that the driver must accept both the lead and lag gap to change lanes. The probability of changing lanes, conditional on the individual-specific term and the direction of change, is given as

$$\begin{aligned} &P\left(l_{nt} = d \,|d_{nt}, \upsilon_n\right) = \\ &P\left(\text{accept lead gap} \,|d_{nt}, \upsilon_n\right) P\left(\text{accept lag gap} \,|d_{nt}, \upsilon_n\right) = \end{aligned} \tag{6.8}$$

where $d_{nt} \in \{\text{Right, Current, Left}\}$ is the direction of change as determined by the target lane choice. l_{nt} is the lane-changing action.

The joint probability density of a combination of target lane (TL) and lane action (l) observed for driver n at time t, and the individual-specific characteristic υ_n is given as

$$P\left(TL_{int}, l_{nt} \,|\upsilon_n\right) = P\left(TL_{int} \,|\upsilon_n\right).P\left(l_{nt} \,\middle|TL_{int}, \upsilon_n\right) \tag{6.9}$$

where $P\left(TL_{int}|\cdot\right)$ and $P\left(l_{nt}|\cdot\right)$ are given by eqs. (6.7) and (6.8), respectively.

Only the driver's lane-changing actions are observed over the sequence of observations. Assuming that, conditional on υ_n, these observations are independent, the joint probability of the sequence of observation for a given driver l_n is given as

$$P\left(l_n \,|\upsilon_n\right) = \prod_{t=1}^{T_n} \sum_{j \in TL} P\left(TL_{jnt}, l_{nt} \,|\upsilon_n\right) \tag{6.10}$$

The unconditional probability of observing the sequence of lane changes by an individual n is obtained by integrating over the distributions of the unobserved individual-specific variables:

$$L_n = P\left(l_n\right) = \int_{\upsilon} P\left(l_n \,|\upsilon_n\right) f\left(\upsilon\right) d\upsilon \tag{6.11}$$

Assuming that the observations from different drivers are independent, the log-likelihood function for all N individuals observed is given by the formula below. The model parameters are estimated by maximizing this function:

$$L = \sum_{n=1}^{N} \ln(L_n) \tag{6.12}$$

The results of applying the methodology are presented through the gap acceptance parameters. The assumption is that the driver evaluates the available adjacent gap in the target lane and decides whether the lane change is possible through the gap acceptance functions. The available lead and lag gaps must be larger than the corresponding critical gaps to be acceptable. The critical gaps (i.e., the smallest gaps a driver is willing to accept) are assumed to follow the log-normal distribution. The mean of the distribution is a function of explanatory variables:

$$\ln(G_{lnt}^{\text{lead cr}}) = X_{lnt}^{\text{lead}} \beta^{\text{lead}} + \alpha^{\text{lead}} \upsilon_n + \varepsilon_{lnt}^{\text{lead}} \tag{6.13}$$

$$\ln(G_{lnt}^{\text{lag cr}}) = X_{lnt}^{\text{lag}} \beta^{\text{lag}} + \alpha^{\text{lag}} \upsilon_n + \varepsilon_{lnt}^{\text{lag}} \tag{6.14}$$

where $G_{lnt}^{\text{lead cr}}$ and $G_{lnt}^{\text{lag cr}}$ are the lead and lag critical gaps in target lane l, respectively. X_{lnt}^{lead} and X_{lnt}^{lag} are explanatory variables that affect the critical gaps. β^{lead} and β^{lag} are the corresponding parameters. υ_n is an individual-specific latent variable that captures the correlations among decisions made by the same driver over time and choice dimensions. α^{lead} and α^{lag} are the coefficients of this latent variable. $\varepsilon_{lnt}^{\text{lead}}$ and $\varepsilon_{lnt}^{\text{lag}}$ are random terms: $\varepsilon_{lnt}^{\text{lead}}\tilde{N}\left(0,\sigma_{\text{lead}}^2\right)$, $\varepsilon_{lnt}^{\text{lag}}\tilde{N}\left(0,\sigma_{\text{lag}}^2\right)$.

The variables that have a significant impact on the critical gaps are the relative speeds with respect to the lead and lag vehicles. The estimated lead and lag gaps are

$$G_{lnt}^{\text{lead cr}} = \exp\begin{pmatrix} 1.541 - 6.210\,\text{Max}\left(0,\Delta V_{lnt}^{\text{lead}}\right) - \\ -0.130\,\text{Min}\left(0,\Delta V_{lnt}^{\text{lead}}\right) - 0.008\upsilon_n + \varepsilon_{lnt}^{\text{lead}} \end{pmatrix} \tag{6.15}$$

$$G_{lnt}^{\text{lag cr}} = \exp\left(1.426 + 0.640\,\text{Max}\left(0,\Delta V_{lnt}^{\text{lag}}\right) - 0.240\upsilon_n + \varepsilon_{lnt}^{\text{lag}}\right) \tag{6.16}$$

where $\Delta V_{lnt}^{\text{lead}}$ and $\Delta V_{lnt}^{\text{lag}}$ are the relative speeds with respect to the lead and lag vehicles, respectively.

The lead critical gap decreases with the relative lead speed (i.e., it is larger when the subject vehicle is faster relative to the lead vehicle). The effect of the relative speed is strongest when the lead vehicle is faster than the subject. In this case, the lead critical gap quickly diminishes as a function of the speed difference. This shows that drivers perceive very little risk from the lead vehicle when it is getting away from them.

In the gap acceptance model, the lag critical gap increases with the relative lag speed: the faster the lag vehicle relative to the subject, the larger the lag critical gap. In contrast to the lead critical gap, the lag gap does not diminish when the subject is faster. A possible explanation is that drivers maintain a minimum critical lag gap as a safety buffer since their perception of the lag gap, through mirrors, is not as reliable as their perception of the lead gap. Estimated coefficients of the unobserved driver characteristic variable υ_n are negative for lead and lag critical gaps. This is consistent with the interpretation of υ_n as being negatively correlated with aggressive drivers who require smaller gaps for lane changing (for detailed results, see Toledo et al., 2005).

6.5.3 Aggregate Calibration

This section presents a mathematical formulation and solution approaches to the aggregate calibration problem. The methodology is appropriate for the simultaneous calibration of supply and demand parameters and inputs to microscopic traffic simulation models.

Let the time period of interest be divided into intervals $h = 1, 2, \ldots, H$. Let X_h denote the vector of OD flows departing their respective origins during time interval h. Let β be the vector of simulation model parameters (possibly also time varying)

that need to be calibrated together with the OD flows. The calibration problem may then be formulated mathematically in the following optimization framework (Balakrishna, 2006; Balakrishna et al., 2006a):

$$\text{Minimize } z(\mathbf{x}_1, \ldots, \mathbf{x}_H, \boldsymbol{\beta}) = \sum_{h=1}^{H} \left[z_1 \left(\mathbf{M}_h, \mathcal{M}_h \right) + z_2 \left(\mathbf{x}_h, \mathbf{x}_h^{\mathrm{a}} \right) + z_3 \left(\boldsymbol{\beta}, \boldsymbol{\beta}^{\mathrm{a}} \right) \right] \quad (6.17)$$

subject to

$$\left.\begin{array}{l} \mathcal{M}_h = f\left(\mathbf{x}_1, \ldots, \mathbf{x}_h, \boldsymbol{\beta}, G_1, \ldots, G_h\right) \\ l_h^x \le \mathbf{x}_h \le u_h^x \\ l^\beta \le \boldsymbol{\beta} \le u^\beta \end{array}\right\}, \qquad \forall\, h \in \{1, 2, \ldots, H\} \quad (6.18)$$

where $\mathbf{M}_h$ and $\mathcal{M}_h$ are the observed and simulated sensor measurements for interval h; $\mathbf{x}_h^{\mathrm{a}}$ and $\boldsymbol{\beta}^{\mathrm{a}}$ are a priori values corresponding to $\mathbf{x}_h$ and $\boldsymbol{\beta}$; z_1, z_2, and z_3 are goodness-of-fit functions. $F(\ )$, the simulation model, is a function of the unknown OD flows and model parameters $\boldsymbol{\beta}$, and network G_h. The network may vary from time period to time period due to accidents, etc. and hence is presented as time-dependent G_h. l_h^x, l^β, u_h^x, and u^β represent lower and upper bounds on the OD flows and model parameters.

The a priori values can ensure reasonable calibrated estimates. They may be based on the modeler's experience and judgment from past studies, or transferred appropriately from similar studies. This problem formulation introduces the flexibility to incorporate other traffic measurements beyond the standard link counts. For example, speeds, occupancies, or travel times may be used. All model inputs and parameters of interest may be calibrated simultaneously, using all information from the available measurements, without iterating between various parameter subsets.

To solve the resulting large-scale optimization problem, the simultaneous perturbation stochastic approximation (SPSA) algorithm developed by Spall (1998, 1999) has been used. The method performs well computationally and in terms of the solution quality. Further details on the aggregate calibration problem and the solution approaches are presented in Balakrishna et al. (2007a).

The methodology outlined above is illustrated through a case study, based on a network in Lower Westchester County, NY (Fig. 6.9). This network is heavily congested, especially during commute periods. Truck traffic is prohibited from parkways. Given the significant truck percentage in the network traffic, passenger cars and truck demand were treated independently by calibrating multi-class demand matrices.

The network representation of the study area comprises 1767 directed links and 482 OD pairs. The data for the calibration process included count data from 33 sensors shown in Fig. 6.9 and an all-day static OD matrix. Disaggregate data on individual vehicles passing through toll plazas were also available. These observations also contained vehicle class information. A time-dependent OD matrix

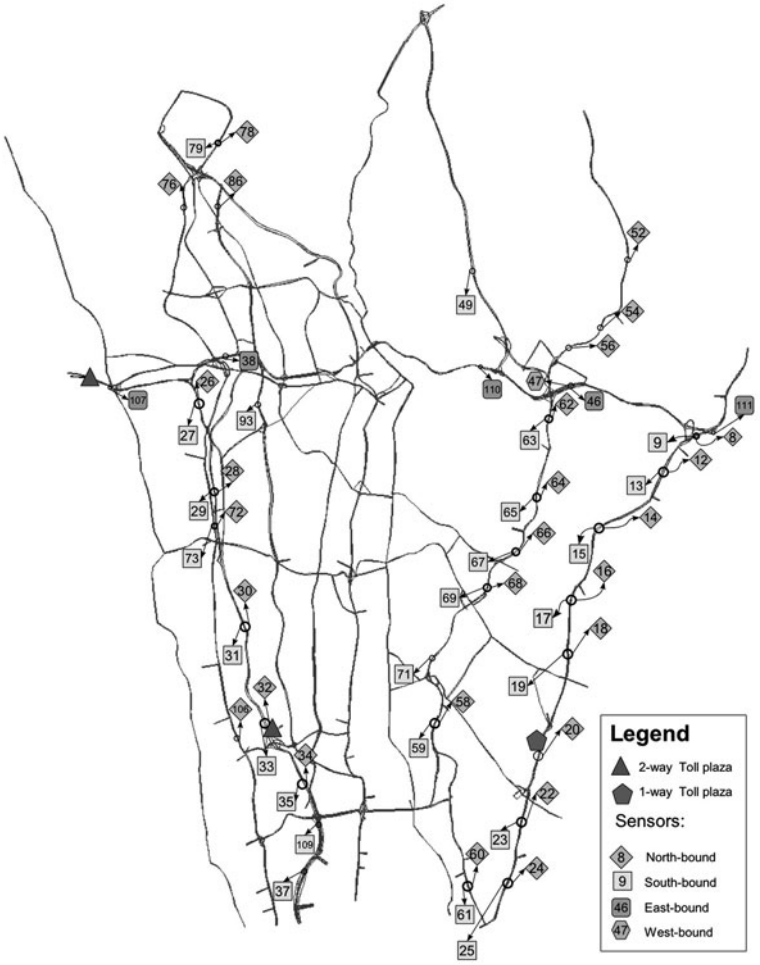


Fig. 6.9 The case study network showing sensor locations used for calibration

was estimated for all vehicles using the SPSA algorithm. This demand was further decomposed into two components (passenger cars and trucks) prior to being input into MITSIMLab using the time-dependent observed vehicle mix from the toll-plaza data. The normalized root mean squared error, root mean squared percent error (RMSPE), mean percent error (MPE), and Theil's coefficient were used as goodness-of-fit statistics to evaluate the calibrated model.

The normalized root mean square (RMSN) error and root mean square percent error (RMSPE) quantify the overall error of the simulator. These measures penalize large errors at a higher rate than small errors. The mean percent error (MPE)

statistics indicates the existence of systematic under- or over-prediction in the simulated measurements.

$$\text{RMSNE} = \frac{\sqrt{N \sum_{n=1}^{N} \left(Y_n^{\text{s}} - Y_n^{\text{o}}\right)^2}}{\sum_{n=1}^{N} Y_n^{\text{o}}} \tag{6.19}$$

$$\text{RMSPE} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} \left[\frac{Y_n^{\text{s}} - Y_n^{\text{o}}}{Y_n^{\text{o}}}\right]^2} \tag{6.20}$$

$$\text{MPE} = \frac{1}{N} \sum_{n=1}^{N} \left[\frac{Y_n^{\text{s}} - Y_n^{\text{o}}}{Y_n^{\text{o}}}\right] \tag{6.21}$$

where N is the number of observations, Y_n^{o} and Y_n^{s} are an observation and the corresponding simulated value, respectively.

Theil's inequality coefficient is a measure of relative error given as

$$U = \frac{\sqrt{\frac{1}{N} \sum_{n=1}^{N} \left(Y_n^{\text{s}} - Y_n^{\text{o}}\right)^2}}{\sqrt{\frac{1}{N} \sum_{n=1}^{N} \left(Y_n^{\text{s}}\right)^2} + \sqrt{\frac{1}{N} \sum_{n=1}^{N} \left(Y_n^{\text{o}}\right)^2}} \tag{6.22}$$

U is bounded between 0 and 1 (where $U = 0$ implies perfect fit between observed and simulated measurements). Theil's inequality coefficient may be decomposed into three proportions of inequality, the bias (U^{M}), the variance (U^{S}), and the covariance (U^{C}) proportions (their sum is equal to 1):

$$U^{\text{M}} = \frac{\left(\bar{Y}^{\text{s}} - \bar{Y}^{\text{o}}\right)^2}{\frac{1}{N} \sum_{n=1}^{N} \left(Y_n^{\text{s}} - Y_n^{\text{o}}\right)^2} \tag{6.23}$$

$$U^{\text{S}} = \frac{\left(s^{\text{s}} - s^{\text{o}}\right)^2}{\frac{1}{N} \sum_{n=1}^{N} \left(Y_n^{\text{s}} - Y_n^{\text{o}}\right)^2} \tag{6.24}$$

$$U^{\text{C}} = \frac{2\,(1 - \rho)\, s^{\text{s}} s^{\text{o}}}{\frac{1}{N} \sum_{n=1}^{N} \left(Y_n^{\text{s}} - Y_n^{\text{o}}\right)^2} \tag{6.25}$$

where ρ is the correlation between the two sets of measurements; s^s and s^o are the standard deviations of the average simulated and observed measurements, respectively and $\overline{Y^s}$ and $\overline{Y^o}$ are their expected values.

The bias proportion reflects the systematic error. The variance proportion indicates how well the model replicates the variability in the observed data. These two proportions should be kept as close to 0 as possible. The covariance proportion measures the remaining error and therefore should be close to 1.

Table 6.1 compares the values of the above statistics for the calibrated model to the values from the initial demand case. All measures improved over the initial values, especially the bias measures.

Table 6.1 Goodness-of-fit statistics for the case study network

Statistics	Calibrated model	Model with a priori demand
RMSPE (%)	22.1	41.6
RMSNE (%)	23.1	47.6
MPE (%)	−5.3	−28.9
Theil's coefficient U	0.113	0.264
Bias proportion	0.116	0.461
Variance proportion	0.015	0.020

6.5.4 Validation

MITSIMLab has been validated in a number of studies. This section discusses the results from a validation study in Stockholm, Sweden (Toledo et al., 2003). The study used a mixed urban-freeway network in the Brunnsviken area, north of the CBD, shown in Fig. 6.10. It contains the E4 motorway connecting the northern suburbs to the CBD. A parallel arterial is also included. These routes experience heavy congestion during the AM peak period. Sensor data from May 1999 were used to calibrate MITSIMLab. Similar data were collected a year later for validation. Measurements of point-to-point travel times and queue lengths by probe vehicles and from aerial photography were also available for validation. Sensor and other measurement locations are shown in Fig. 6.10. A static AM peak OD flow matrix, previously developed for planning studies, was used in the OD estimation.

The model validation used the comparison of measured and simulated traffic flows, travel times, and queue lengths during 2 h of the AM peak data from May 2000 at 15 min intervals. The traffic flows were also used in the estimation of OD matrices for this period. Figure 6.11 compares average simulated travel times and individual probe vehicle observations for the inbound section CD, which was the most congested during this period. The section also includes a bus lane for buses and other commercial vehicles. In general, simulated travel times match observed travel times well.

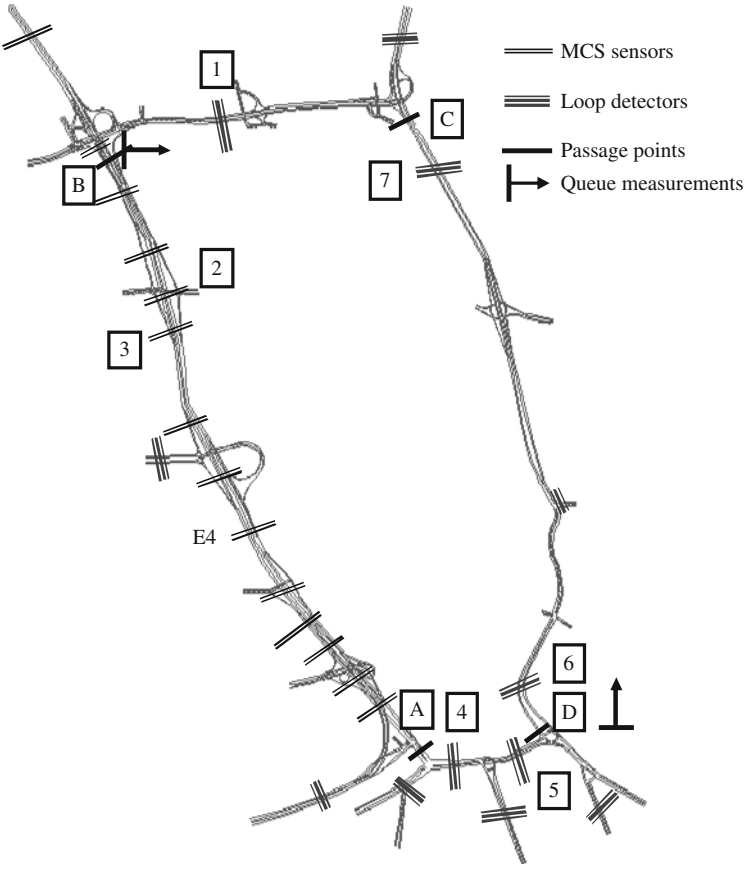


Fig. 6.10 The Brunnsviken network

Similarly, Fig. 6.12 presents the validation results for the queue lengths measured by the probe vehicles and from aerial photographs. Queues are represented in the simulation both by magnitude and time of occurrence.

6.6 Extended Modeling Capabilities: Working with External Applications

MITSIMLab has been integrated with a number of external applications. This section presents two cases: the use of MITSIMLab to evaluate the performance of DynaMIT, a dynamic traffic assignment (DTA), and traffic prediction generation tool; and the integration of MITSIMLab with a mesoscopic traffic simulation model to create a hybrid model.

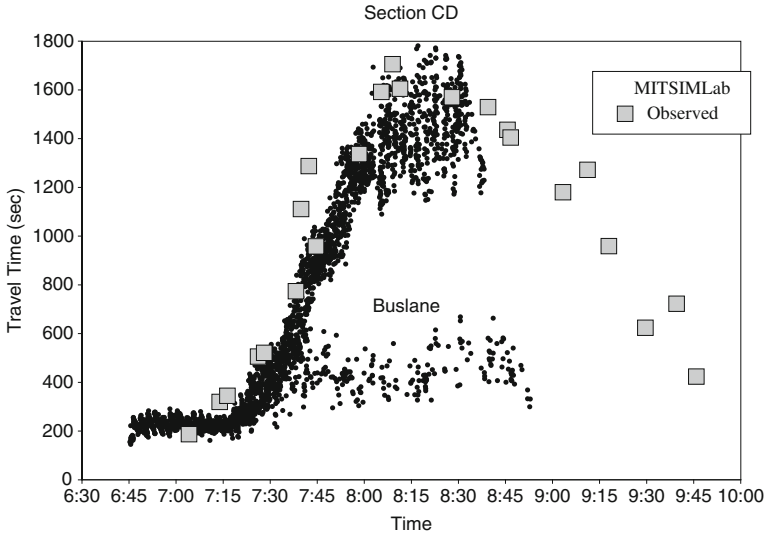


Fig. 6.11 Point-to-point travel time validation results (section CD)

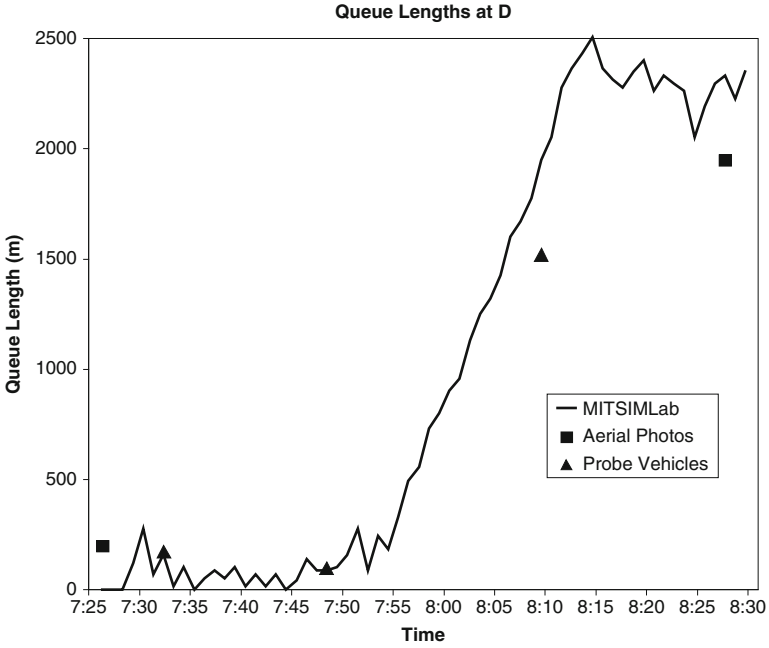


Fig. 6.12 Queue length validation results (location D)

6.6.1 Closed Loop with DynaMIT

MITSIMLab provides a controlled environment to conduct objective evaluations of advanced ITS concepts, such as DTA-based traffic prediction and information generation. MITSIMLab was integrated with DynaMIT (dynamic network assignment for the management of information to travelers) in a closed-loop system (Balakrishna et al., 2005). DynaMIT is a model system for (real-time) traffic estimation and prediction. A detailed description of DynaMIT is provided in Chapter 10.

The closed-loop system provides a framework for off-line evaluation of dynamic traffic management systems such as DynaMIT. In the closed-loop evaluation, MITSIMLab replaces the real world. Figure 6.13 illustrates this evaluation framework and the interactions between the two applications.

The two models are run in parallel using the same network and scenario database. Traffic data from the simulated surveillance system in MITSIMLab are transmitted in real time to DynaMIT. Prediction-based guidance is passed back to the control and routing devices simulated in the TMS and then to equipped drivers in MITSIMLab. The drivers' reactions to the disseminated information and changes in the control system are reflected in subsequent traffic flows, which are measured by the simulated surveillance system and transferred to DynaMIT for the next prediction generation step. Network performance measures are computed to assess the effectiveness of the guidance and dissemination system. The integration of DynaMIT within TMS is similar to the interface between DynaMIT and a real traffic control center.

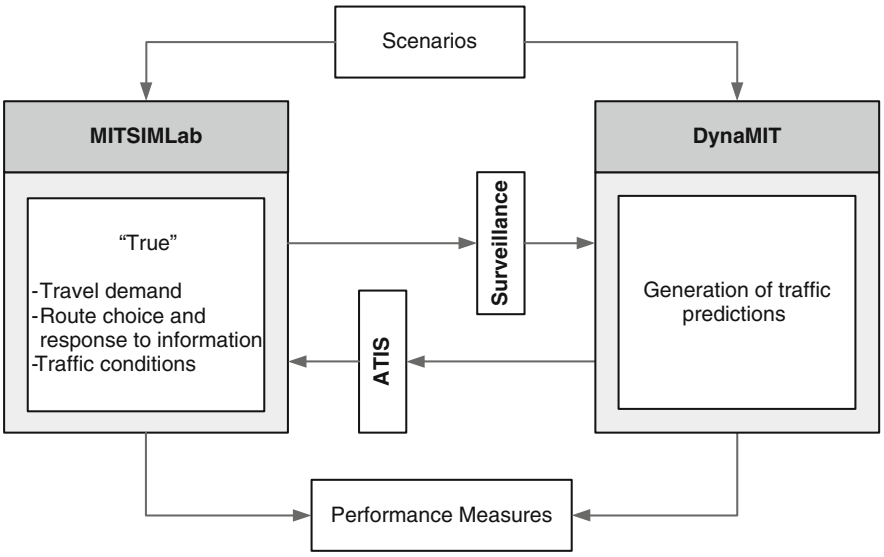


Fig. 6.13 MITSIMLab–DynaMIT closed-loop evaluation framework

The advantage of the closed-loop laboratory is that it allows great flexibility in performance evaluations. The testing of advanced traveler information systems in response to various parameters and design characteristics is an example:

- *Modeling errors.* DynaMIT uses a number of models to simulate the demand aspects of the transportation system (e.g., route choice, departure times) and network performance (e.g., queue formation and dissipation). MITSIMLab also uses OD flows and travel behavior models (i.e., route choice). The error associated with the models used by DynaMIT (compared to the "true" behavior in MITSIMLab) can be controlled, and its impact on the effectiveness of the system assessed.
- *Design parameters*. A number of design parameters influence the effectiveness of the system. Examples of such parameters of interest include the prediction horizon, the frequency of updating the traffic information, and the time resolution of the provided guidance.
- *Computational delay*. Many strategies are computationally demanding. The time to generate a new strategy for implementation depends on the size of the network and the available computational resources. The laboratory tests the effectiveness of the system as a function of the computational delay.
- *Design of the surveillance system*. The impact of the location, type, and number of sensors can be assessed. In addition, sensors are assigned an (measurement) error attribute, allowing for the evaluation of the surveillance system characteristics. The impact of the accuracy of information with respect to incidents and their severity on the effectiveness of the system can also be evaluated. Typically, incident information may be delayed and duration uncertain.
- *Communication system and interfaces.* Important aspects of the communications between the various elements of the system can be modeled and their significance assessed. Such parameters include latency in information transmission and errors and noise in the information.

6.6.2 Hybrid Simulation

Microscopic simulation models provide a detailed representation of the traffic process. Other types of traffic simulation models, namely macroscopic and mesoscopic, capture traffic dynamics in less detail. But they require less input data preparation and can simulate large-scale networks efficiently (from a computational point of view). Hybrid simulations combine mesoscopic or macroscopic models for most of the network and microscopic models in the areas of interest. Hybrid models have the advantages of both types of simulation since they combine high fidelity microsimulation in areas of particular interest, with mesosimulation of the surrounding areas (in order to represent routing decisions more accurately). Another advantage of the integration is that it reduces the computational requirements and the data collection and calibration effort of the overall model.

An important aspect in developing a hybrid meso–microtraffic simulation model is the identification and implementation of conditions for consistent interfaces between the two components. These conditions range from structural compatibility issues in terms of modeling traffic flows in the two models, to consistency of traffic dynamics at the meso–microboundaries, to compatibility of route choice (see Burghout et al., 2005).

Burghout (2004) integrated MITSIMLab with Mezzo, a mesoscopic model, to illustrate the principles of integration and its advantages. The hybrid simulation model was demonstrated through its application to the Brunnsviken network in Stockholm. The network was divided into a mesoscopic part in the north, which consists mainly of freeways, and a microscopic part in the south, which consists of complex intersections with coordinated signal control and a large roundabout as shown in Fig. 6.14.

Table 6.2 summarizes the fit of the simulated flows in the hybrid model to field observations and compares it against stand-alone applications of MITSIMLab and Mezzo. The RMSPE and Theil's coefficient statistics indicate that MITSIMLab provides the best fit. But the hybrid simulation outperforms the mesoscopic model and suffers only a slight reduction in fit. Using the hybrid model improves the fit compared to the mesoscopic one, in the microscopic part of the network and in the mesoscopic part. Using Theil's proportions to break down the error shows that the mesoscopic model has a larger systematic error compared to the hybrid model. The computational time for the hybrid model is also superior. This difference is expected to increase as the network grows.

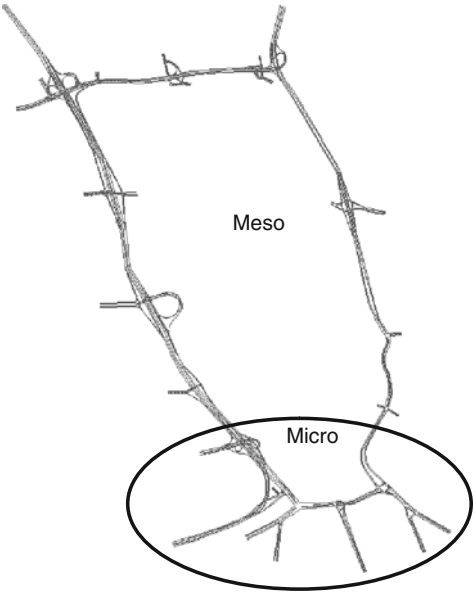


Fig. 6.14 Hybrid model for the Brunnsviken network

Table 6.2 Results of various models

Statistics	MITSIMLab	Mezzo	Hybrid
RMSPE (%)			
Entire network	12	16	15
Meso part	10	13	11
Micro part	14	18	17
Theil's coefficient U	0.051	0.055	0.054
Bias proportion	0.001	0.147	0.075
Variance proportion	0.010	0.002	0.017

6.7 Advanced Case Studies and Applications

6.7.1 ATIS Evaluation and Design

The closed-loop system from Section 6.6.1 was used to evaluate several design aspects of information generation systems. The case study detailed in Balakrishna et al. (2005) explores the impacts of several factors including the guidance penetration rate (i.e., fraction of drivers with access to the information), the frequency of information update, and errors in the quality and effectiveness of travel time guidance generated by the demand prediction and route choice model. The case study uses the Central Artery/Tunnel network in Boston, shown in Fig. 6.15. This network consists of 182 nodes and 211 links.

The case study included the AM peak period starting at 7:00 AM. At 7:10 AM, an incident occurred in the Ted Williams Tunnel, blocking one lane and reducing the speed on the other lane. The incident lasted 20 min. Approximately 3500 vehicles per hour flow through the Ted Williams Tunnel. The simulated incident created substantial delays to travelers. The simulation lasted until 8:45 AM to ensure that traffic conditions were returned to normal after the end of the incident.

Guided drivers are assumed to have access to descriptive information in their vehicles. Various values of the percentage of guided vehicles (0, 20, 30, 50, 70, and 100%) were tested. The results are summarized in Fig. 6.16. The results indicate decreasing average travel times as the percentage of guided drivers increases. Some over-reaction was indicated by the slight increase in travel times as the guided fraction increased beyond 70%. Predictive guidance does not eliminate over-reaction due to the discrete nature of the representation of the problem, as well as modeling and algorithmic approximations. For example, the results indicate that when the update frequency decreases, the shorter update intervals allow the system to quickly adjust to changing network conditions, and the impact of over-reaction is almost eliminated. This result highlights the need for better, more accurate anticipatory traveler information that accounts for future demands and driver behavior.

For this case study, MITSIMLab and DynaMIT use the same OD matrices and route choice model based on the path-size logit structure to represent travel

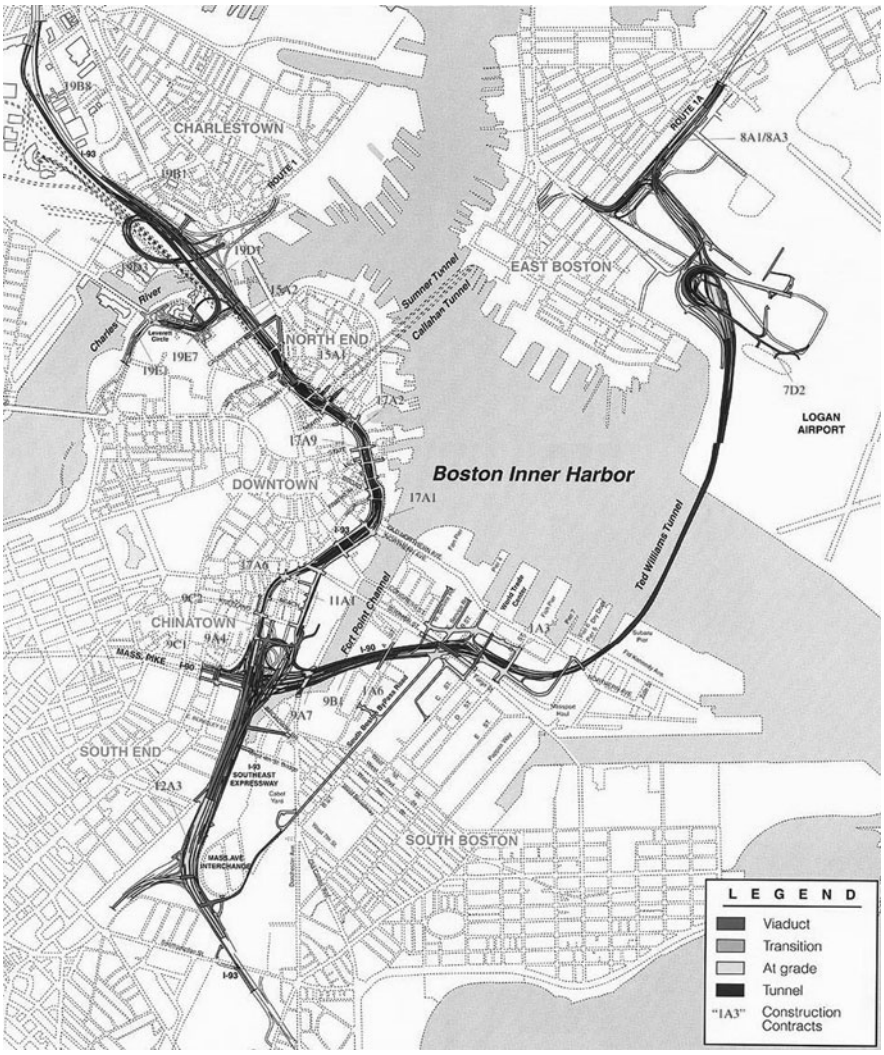


Fig. 6.15 The Central Artery/Tunnel network (source: http://www.masspike.com/bigdig/multimedia/plans.html)

demand. It is unrealistic to expect guidance generation models to perfectly estimate demand and predict route choices. The impact of errors in these factors was assessed by introducing errors in the predicted OD flows. Travel time coefficients used in DynaMIT's route choice model were also modified to include an error relative to the "true" value used in MITSIMLab. Table 6.3 summarizes the average travel times in the network for the different scenarios. The results indicate that the effectiveness of the system, as measured by the average travel times, is influenced by the demand and route choice prediction errors used in DynaMIT. The combined

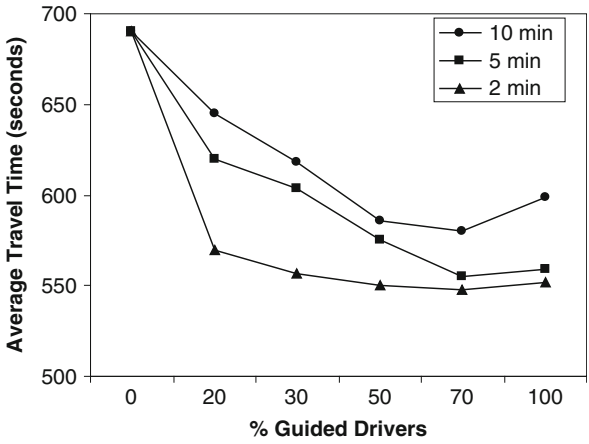


Fig. 6.16 Effect of guidance penetration rate

Table 6.3 Effect of errors in the demand on the average travel times (s)

	Route choice parameter error		
OD prediction error	0	−20%	+20%
0	618	620	627
+20%	625	633	634

effect of error in the OD matrix and the route choice model is greater than the sum of the effects of the two individual sources of error. Differences between these results and those from previous studies may reflect the network specifics, demand characteristics, assumptions of the ATIS design, and the overall structure of the evaluation methodology. The differences also underline the importance of the simulation-based laboratory for detailed evaluations.

6.7.2 Evaluation of Advanced Signal Priority Strategies

This case study evaluates bus operations through various conditional signal priority strategies for a bus rapid transit (BRT) line in an urban network in Stockholm, Sweden. The time period of interest is 7:30–8:30 AM. The BRT routes are served by articulated buses equipped with GPS-based AVL systems. The study area is shown in Fig. 6.17. Traffic in the side streets crossing the three arterials is relatively low compared to traffic on the three arterials. There are seven signalized intersections in the study area. One of them is a signalized pedestrian and bicycle crossing. Three local lines and one BRT line operate in this section. Local buses have 15-min headways during the peak periods, and the BRT-articulated buses operate with 7.5-min headways. The local and BRT services share the bus stops.

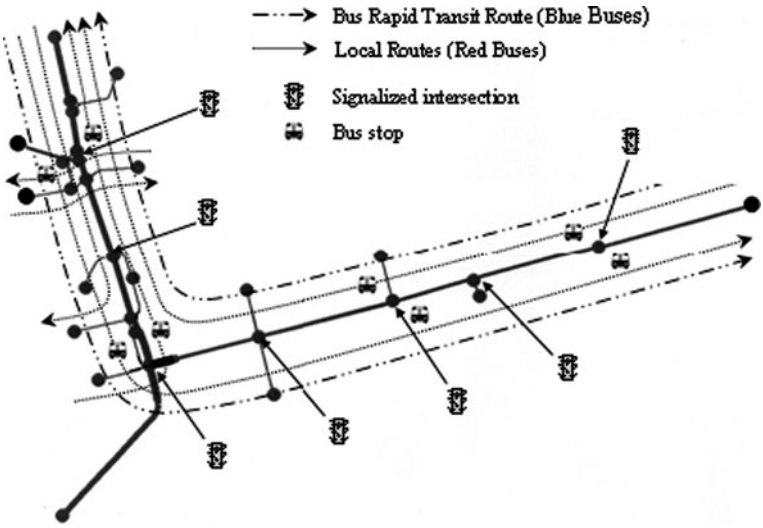


Fig. 6.17 Study network showing bus routes, stops, and signalized intersections

The purpose of the case study is to evaluate the extent to which transit signal priorities improve the performance of the bus lines and to assess its impact on the general traffic in the section. Four priority implementations are compared: (1) no priority (base case), (2) unconditional priority, (3) conditional priority only for buses with more than 30 passengers, and (4) conditional priority only for buses with headways that exceed 7.5 min. A sensitivity analysis that explores the effects of increased side street demand is also conducted.

Figure 6.18 summarizes the main results. The top part of the figure shows the results for the base-case demand. The average vehicle travel times are shown for different groups of vehicles: all vehicles, BRT, and vehicles crossing the section from the side streets. Average BRT travel times decreased as the priority conditions became less restrictive. The lowest average travel times occurred under unconditional priority. The signal priority reduced the variability of BRT travel times. The conditional priority strategies can achieve similar BRT travel times compared to the unconditional priority without granting priority quite as often. With the relatively low base-case demand, the change in average travel time for side-street vehicles is small. It does not support general conclusions about the trade-offs between transit travel time savings and side-street travel time penalties. Additional simulations were run with a 40% increase in side-street demand. The results are shown in the bottom part of Fig. 6.18. The load-based conditional priority yields BRT travel times that are similar to the BRT travel times under unconditional priority. The improvements in BRT travel times under conditional priority have considerably lower adverse impacts on side-street traffic.

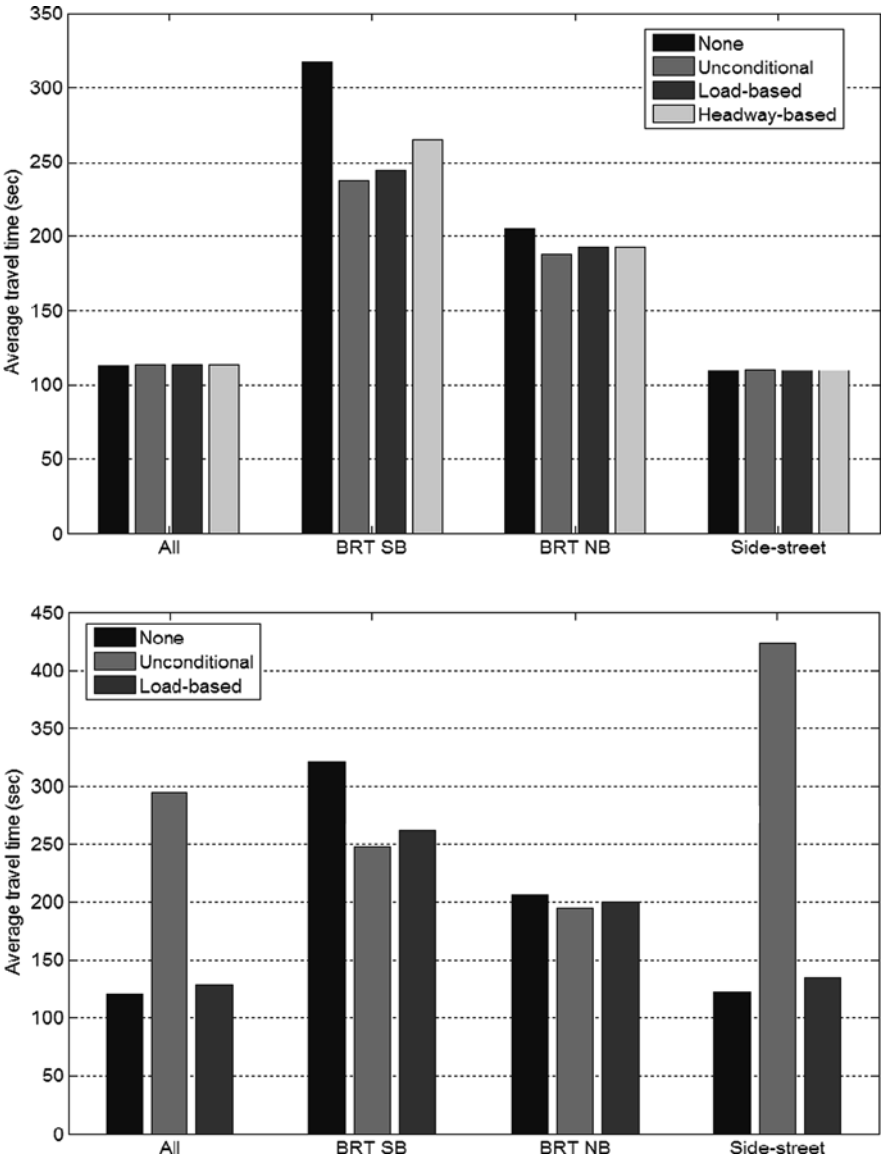


Fig. 6.18 Signal priority impact on travel times: base case (*top*), increased side-street demand (*bottom*)

6.8 Advanced Modeling Details

In recent years, the driving behavior models in MITSIMLab have improved in their fidelity to freeway and urban sections under congested conditions. This section

presents two developments: the freeway merging model, and the lane-choice and lane-changing models for urban arterials.

6.8.1 Freeway Merging Model

This application deals with drivers' merging behavior when entering freeways. Traditional merging models are based on gap acceptance, i.e., drivers merge when an acceptable gap is available. However, in congested traffic, acceptable gaps are often unavailable and more complex merging phenomena are observed. Drivers may merge through courtesy of the lag driver in the target lane or become impatient and decide to force merge, compelling the lag driver to slow down.

To capture this behavior, drivers' selection of a merging tactic needs to be included in the model. The decision framework is presented in Fig. 6.19. At each time interval, drivers select a merging plan (tactic) and decide whether they can use this tactic to merge. Critical gaps depend on the chosen plan. The merging plan may evolve dynamically with changing conditions. For example, a driver may initially try to merge normally. But as the driver approaches the end of the merging lane, he may decide to force merge. The probabilities of transitioning between plans are affected by the risk associated with the merge, the characteristics of the driver such as patience level, urgency, and aggressiveness as well as inertia to continue the previously chosen merging tactic (state dependence). These effects are captured by variables such as relative speed and acceleration of the mainline vehicles,

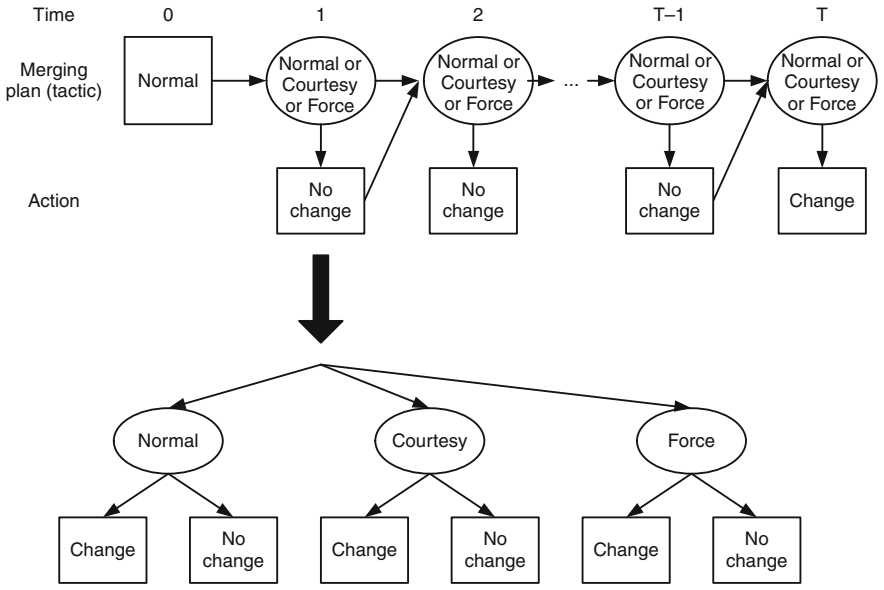


Fig. 6.19 Framework of the merging model

delay associated with the merge, density of traffic, and distance from the end of the merging lane.

The parameters of the model were estimated with trajectory data collected from I-80 in California using the maximum likelihood method. In the trajectory data, only the final execution of the merge is observed. The sequences of tactics drivers applied are unobserved. A hidden Markov model formulation is used to model these latent tactics. Estimation results showed that the inclusion of the three merging tactics and the differences in critical gaps associated is justified by the data. The final results showed that drivers are willing to accept smaller lead and lag gaps if they perceive that the lag vehicle is courtesy yielding.

To demonstrate the benefits of including latent plans, the model described above was compared against a reduced model that does not incorporate latent tactics. In this model, instantaneous single-level gap acceptance was used. The latent plan model showed a significantly better goodness of fit in statistical tests. Both models were implemented in MITSIMLab for evaluation using data from a section of US 101 in California. The validation results for the location of merges are presented in Fig. 6.20. The latent plan model more realistically replicated the observations on lane-specific flows, speeds, and the locations of merges. The detailed model structure, estimation, and validation results are presented in Choudhury et al. (2007).

6.8.2 Arterial Lane-Changing Model

MITSIMLab has been extended to incorporate a number of integrated driving behavior models appropriate for urban streets. Arterial corridors exhibit a set of varied driving activities that differ by lane and location. These activities encompass trip destination activities (e.g., parking, entering transit stops, right turns, left turns), trip origination activities (e.g., exiting a parking spot, exiting transit stops), and

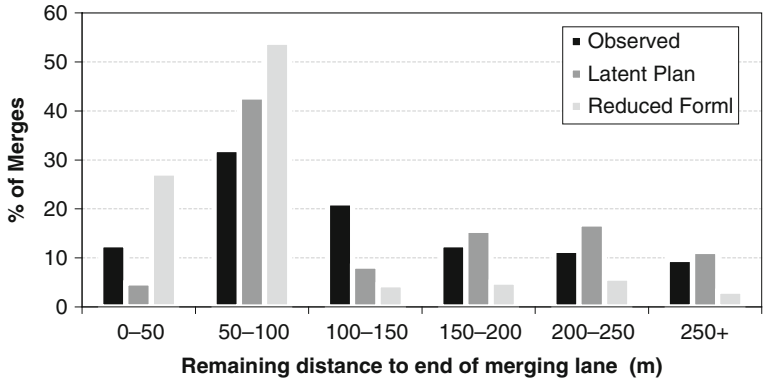


Fig. 6.20 Comparison of location of merges

complex routing behaviors (e.g., permissive left turns, pedestrian-impeded right turns). Drivers familiar with the network may be aware of these activities and their likely locations. They often make appropriate tactical lane positioning decisions to minimize their travel times and driving efforts. The "look-ahead" or "plan-ahead" distances (i.e., how far downstream do drivers "see" in advance) can vary significantly among drivers depending on their individual traits (e.g., planning capability) as well as their experience and familiarity with the network.

This look-ahead distance and the associated heterogeneity can substantially affect lane-changing behavior in urban arterials and was explicitly accounted for in the arterial lane-changing model. The model also captures the time required to complete the lane change (the time elapsed from the instant an adjacent gap is found to be acceptable to the instant the driver physically moves to the new lane) by introducing an additional level that captures the decision to execute/complete the lane change (Fig. 6.21). This shows that although a gap is acceptable, the actual execution of the lane change can depend on different factors (e.g., type and speed of the vehicle, trend of the change in gap size).

The parameters of the model were estimated with trajectory data collected from Lankershim Boulevard in Los Angeles using the maximum likelihood method. Estimation results showed that the path-plan considerations, inertia effect, and lane attributes (e.g., queue-ahead variable, in particular) are pre-dominant factors behind arterial lane-changing decisions as opposed to neighborhood conditions (speed and spacing of adjacent vehicles). The driver's look-ahead distance is normally distributed between 50 and 500 m. In the lane-changing execution model, the results showed that drivers tend to execute the lane change faster if the speed of the subject vehicle is high and/or if the corresponding adjacent gap is reducing (the lag vehicle is faster than the lead vehicle). A comparison of estimation results indicates that addressing the heterogeneity in plan-ahead distances and the execution of the lane change significantly improves the fit to the observations.

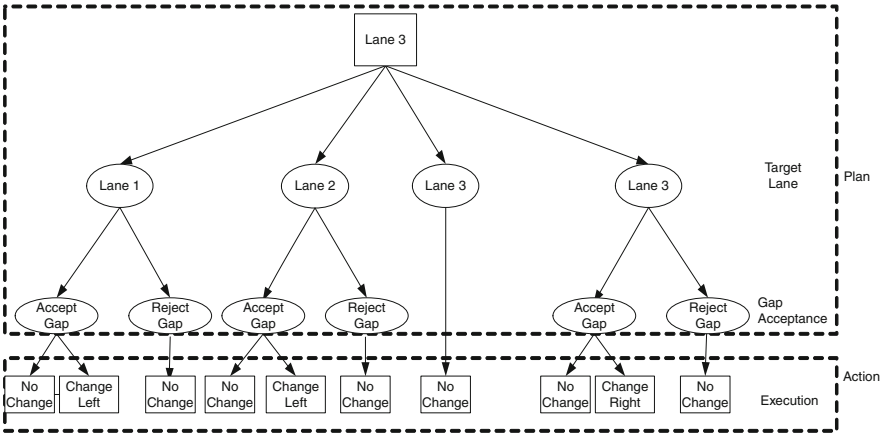


Fig. 6.21 Arterial lane-changing model

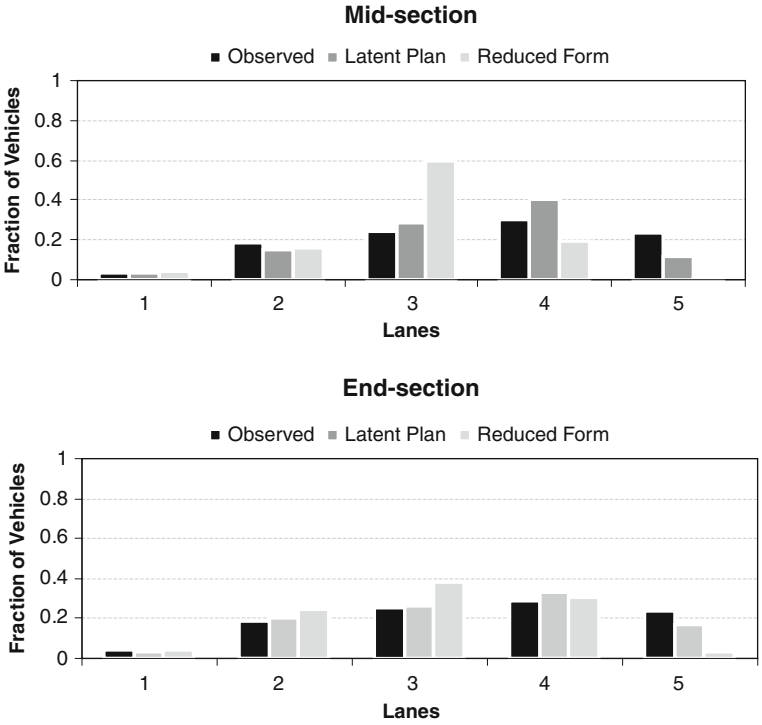


Fig. 6.22 Comparison of vehicle lane distributions

This was further strengthened by a validation case study within MITSIMLab, where the simulation outputs of the urban lane selection model were compared with the MITSIMLab lane changing for freeway traffic models. The results indicated a significant improvement in replicating vehicle distribution among lanes, at mid-sections in particular. The comparison of lane distributions at mid- and end sections obtained from each of the models is presented in Fig. 6.22. The detailed model structure, estimation, and validation results are presented in Choudhury et al. (2008).

References

Ahmed KI (1999) Modelling drivers' acceleration and lane changing behaviors. Ph.D. thesis, Massachusetts Institute of Technology.

Balakrishna R (2006) Off-line calibration of dynamic traffic assignment models. Ph.D. thesis, Massachusetts Institute of Technology

Balakrishna R, Koutsopoulos HN, Ben-Akiva M, Fernandez-Ruiz BM, Mehta M (2005) Simulation-based evaluation of advanced traveler information systems. Transport Res Rec J Transport Res Board 1910:90–98

Balakrishna R, Ben-Akiva M, Koutsopoulos HN (2006a) Time-dependent origin–destination estimation without assignment matrices. Proceedings of the 2nd international symposium on transport simulation, Lausanne, Switzerland

Balakrishna R, Koutsopoulos HN, Ben-Akiva M (2006b) Simultaneous off-line demand and supply calibration of dynamic traffic assignment systems. Proceedings of the 85th annual meeting of the transportation research board, Washington, DC

Balakrishna R, Antoniou C, Ben-Akiva M, Koutsopoulos HN, Wen Y (2007a) Calibration of microscopic traffic simulation models: methods and application. Transportation Res Rec J Transport Res Board 1999:198–207

Balakrishna R, Ben-Akiva M, Koutsopoulos HN (2007b) Off-line calibration of dynamic traffic assignment: simultaneous demand–supply estimation. Proceedings of the 86th annual meeting of the transportation research board, Washington, DC

Ben-Akiva M, Bierlaire M (2003) Discrete choice models with applications to departure time and route choice. In: Hall R (ed) Handbook of transportation science, 2nd edn. Kluwer Academic, Boston, pp. 7–38

Burghout W (2004) Hybrid microscopic–mesoscopic traffic simulation. Ph.D. thesis, Royal Institute of Technology, Stockholm, Sweden

Burghout W, Koutsopoulos HN, Andreasson I (2005) Hybrid mesoscopic–microscopic traffic simulation. Transport Res Rec (1934):218–225

Choudhury C (2007) Modeling driving decisions with latent plans. Ph.D. thesis, Massachusetts Institute of Technology

Choudhury C, Ben-Akiva M, Toledo T, Lee G, Rao A (2007) Modeling cooperative lane-changing and forced merging behavior. Proceedings of the 86th transportation research board annual meeting, Washington, DC

Choudhury C, Ramanujam V, Ben-Akiva M (2008) A lane changing model for urban arterials. Proceedings of the 3rd international symposium of transport simulation, Gold Coast, Australia

Gazis D, Herman R, Rothery R (1961) Nonlinear follow-the-leader models of traffic flow. Oper Res 9:545–567

Sheffi Y, Powell WB (1982) An algorithm for the equilibrium assignment problem with random link times. Networks 12:191–207

Spall JC (1998) Implementation of the simultaneous perturbation algorithm for stochastic approximation. IEEE Trans Aerospace Electron Syst 34:817–823

Spall JC (1999) Stochastic optimization, stochastic approximation and simulated annealing. In: Webster JG (ed) Wiley encyclopedia of electrical and electronics engineering. Wiley, New York, pp 529–542

Toledo T (2008) Integrated model of driving behavior. VDM Verlag Dr. Müller, Saarbrucken, Germany

Toledo T, Koutsopoulos HN, Davol A, Ben-Akiva M, Burghout W, Andreasson I, Johansson T, Lundin C (2003) Calibration and validation of microscopic traffic simulation tools: Stockholm case study. Transport Res Rec 1831:65–75

Toledo T, Ben-Akiva M, Darda D, Jha M, Koutsopoulos HN (2004) Calibration of microscopic traffic simulation models with aggregate data. Transport Res Rec J Transport Res Board 1876:10–19

Toledo T, Choudhury C, Ben-Akiva M (2005) A lane-changing model with explicit target lane choice. Transport Res Rec J Transport Res Board 1934:157–165

Yang Q (1997) A simulation laboratory for dynamic traffic management systems. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA

Yang Q, Koutsopoulos HN (1996) A microscopic simulator for evaluation of dynamic traffic management systems. Transport Res C 4(3):113–129

Yang Q, Koutsopoulos HN, Ben-Akiva M (2000) A simulation laboratory for evaluating dynamic traffic management systems. Transport Res Rec 1710:122–130

Chapter 7
Traffic Simulation with SUMO – Simulation of Urban Mobility

Daniel Krajzewicz

7.1 Introduction

"*S*imulation of *U*rban *MO*bility" (Krajzewicz et al., 2002; Krajzewicz et al., 2004; SUMO, 2001–2009), or "SUMO" for short, is a microscopic road traffic simulation. The work on SUMO's design began in the year 2000, with the first implementation being started in the year 2001. In the beginning, SUMO was developed in collaboration between the Center for Applied Informatics Cologne (ZAIK) and the Institute of Transportation Systems (ITS), at the German Aerospace Center (DLR). Since 2004, the work on SUMO is continued at the DLR only, though with contribution from external organizations or individuals.

SUMO is available as "open source" under the GNU General Public License (GPL, 2009), both as source code and in compiled, executable form for multiple Windows and Linux platforms. The reason for building an open-source traffic simulation was twofold. While working on traffic simulations within the academic field, it was noted that many different, small simulations were developed as tools within diploma or doctoral theses, in order to evaluate the objective that was the thesis' real topic. Often, these simulations were incomplete due to the large amount of problems that must be solved for having a complete traffic simulation, and after the thesis was completed, the used simulations were not made available to the public – they disappeared. So on the one hand, the work to be done to have a proper traffic scenario being simulated was repeatedly redone, wasting resources and distracting from the actual topic of interest. On the other hand, the results of such scientific work are hardly comparable as long as the simulations used to evaluate them have different features, use different (sub-) models and implementation details. Because one of the main research areas of the Institute of Transportation Systems was to compare and evaluate different models and algorithms related to traffic and traffic management, the idea was created to develop a free, extensible traffic simulation, so that

D. Krajzewicz (✉)
Institut für Verkehrssystemtechnik, Deutsches Zentrum für Luft- und Raumfahrt e.V., Rutherfordstr. 2, 12489 Berlin, Germany
e-mail: daniel.krajzewicz@dlr.de

J. Barceló (ed.), *Fundamentals of Traffic Simulation*,
International Series in Operations Research & Management Science 145,
DOI 10.1007/978-1-4419-6142-6_7,

(a) different people may use it as a base test bed for their own applications and (b) these applications would get more comparable.

This idea and the academic context SUMO is developed within highly influenced design criteria and SUMO's features. As main design criteria, the following were stated:

- Portability
  The simulation must run on any common environment, because many research organizations were using Linux or – at the time the work on SUMO started – Solaris as operating system.
- Extensibility
  The simulation must be open and easy to understand, so that someone who has not contributed to the original development can adapt it fast to his or her own needs.

In addition, the first application SUMO was meant to be used for was the simulation of traffic based on the daily activities of a synthetic population of the city of Cologne. This synthetic population – together with their activities – was based upon daily activity reports collected during the "Zeitbudgeterhebung 1992" project (Blanke et al., 1996). Each modeled person has certain mobility wishes and adapts his/her behavior regarding these and the possibilities to accomplish them. A common example is a person who has to re-plan his/her afternoon activities due to a jam which made him/her come too late to work. The process of generating this population is described in Hertkorn and Kracht (2002), Hertkorn and Wagner (2004), and Hertkorn (2004). For applying SUMO within this research, additional features of the software were needed:

- Small Memory Footprint
  The simulation must be able to handle scenarios covering large city areas on a standard work station.
- Execution Speed
  Simulation of large areas must be performed as fast as possible.

These quality requests strongly influenced and still influence SUMO's design and implementation, but they are only the first set of conditions that shape SUMO's development. The second is that SUMO has been developed continuously for eight years now, but only in the frame of the projects the Institute of Transportation Systems or external contributors are working at. During this time, SUMO had been improved significantly, but only along the features needed by the actually done projects. For short, a road map for building the best traffic simulation ever does not exist. As a result, the simulation was "incomplete" during the first years (and still is), simply because some features one would expect were not needed for the current research.

The third influence is the fact that most of the users are not traffic scientists but computer scientists. There may be several reasons for this. At first, SUMO is not an established or even certified traffic simulation as others may be. Furthermore, a user

must have a high affinity toward computer systems, meaning that she/he should be able to work on the command line with no graphical support, and, in some cases, even write own programs for preparing the inputs and evaluating the outputs. Also, we assume that traffic scientists, consultants working in this area, or traffic administrations already possess a traffic simulation and/or are willing to spend money on an established system. This is often not the case if a single diploma thesis in computer science shall be written. The lack of users with a traffic science background has a strong effect on the feedback we get. Only few questions arise about the used models or are in any means related to research on traffic simulations as such.

Nonetheless, the aimed quality criteria could be achieved and we were able to use SUMO successfully in several projects over the past years. The addressed questions range over a set of traffic management aspects, from evaluating new adaptive traffic lights for single intersection control to monitoring and forecasting traffic within large-scale areas. Possible usages will be briefly described by examples in the following.

Being a major use case for traffic simulations, optimization of traffic lights was also addressed within two of our projects, though with two different granularities. Within the project "OIS" – "optical information systems" – a new control algorithm for an intersection was developed and evaluated (Krajzewicz D et al., 2005). This algorithm uses information gained by recognizing vehicles within images taken via cameras located at the controlled intersection. Using this recognition, the system is capable to compute the queue lengths in front of the traffic lights and the implemented algorithm uses these lengths to decide which stream should get green for a longer time. The simulation's task was to evaluate this algorithm's performance in comparison to the real world's traffic light system applied at the simulated junctions

Traffic light systems had also been investigated within the project "ORINOKO," a project performed within the German research initiative "Verkehrsmanagement 2010" (Traffic management 2010, 2009). Here, SUMO was used for evaluation of the new weekly signal program plan developed by one of the project partners. This plan is used within a large area around Nürnberg's fair trade center, the "VLS" area. In order to predict the new plan's performance, a very detailed representation of the area was necessary, including both a very detailed network a and demand representation. The network was modeled using NAVTEQ database (2009), and supplemented by number of lanes, design speeds, intersection geometry, and traffic light information. The demand had been computed by a calibration of the VALIDATE (2009) demand data set for this area with available induction loop count data. Additionally, SUMO was extended to simulate weekly program signal plans. This included the implementation of two methods for switching between different programs, named GSP and Stretch.

Due to the relatively small size of the investigated networks, the mentioned projects are not representative for the intended usage of SUMO. The simulation of much larger networks has been performed within the projects "INVENT" and "WJT"/"Soccer"/"DELPHI." While the latter complex of projects is described more deeply in Section 7.7, "INVENT" will be shortly presented now; it shows the

intended major application of SUMO: the evaluation of large-scale impacts of traffic management strategies or new technologies.

An example of such evaluations done in "INVENT" (2009) was to determine the effects of different vehicle routing strategies in the case of recognized jams, such as ones caused by oversaturation or temporal access restrictions to parts of the city. Here, the simulation was coupled to a routing module via a socket connection for allowing the simulation to send current travel times to the router and the router to send new routes for the simulated vehicles. The routing module was extended to allow the specification of certain modalities for rerouting vehicles, affecting the frequency of route recomputation (either each x s or in the case of an event), and the information available during the computation of a new route (complete knowledge, knowledge about large delays, additional weights given by a user, road closures). The evaluations were done for large-scale networks for two German cities, Magdeburg and Munich. The initial scenarios were set up by the PTV AG and provided in the form of VISUM and Vissim scenarios. For Munich only the northern highway network was simulated. The Magdeburg scenario consists of a road network which includes major roads and a demand of about 600,000 vehicles for a complete day. SUMO's ability to deal with large scenarios becomes visible when knowing that a complete day's simulation of this scenario needs about 45 min using recent PCs.

Another project from the "Verkehrsmanagement 2010" research initiative called "TrafficOnline" (Ehrenpfordt et al., 2006; Höpfner et al., 2006; TrafficOnline, 2009) gave us the opportunity to use SUMO as a platform for evaluation of a GSM-based traffic surveillance system. The simulation's duty was to reproduce in-vehicle telephony behavior within normal situations for different road network types, such as highways or inner-city areas. Again, the scenarios were set up using networks from a NAVTEQ database and induction loop measures and both the obtained traffic flows and the simulated GSM talk statistics were compared with values from the real world. After their validation, these base scenarios were extended by other influences, such as traffic jams, or other transport modes running parallel to the normal traffic, such as city rail, fast rail, or explicit bus lanes. The purpose was to evaluate (a) the quality of the surveillance under undisturbed conditions, (b) the reaction time of the system in the case of incidents, and (c) how well the surveillance system behaves under different conditions in terms of disturbances by other sources of GSM talks. The evaluations were done by sending the GSM talk statistics gained from the simulation runs to the Institute of Transportation and Urban Engineering (IVS) of the Technical University of Braunschweig, which was the developer of the surveillance system. The IVS used these statistics to predict mean velocities on the road network for intervals of 5 and 15 min. These were sent back to DLR and compared against the travel times obtained from the according simulation.

Two recent projects of external contributors should be mentioned because both built upon own extensions of the simulation module. Honomichl (2008) has investigated attacks on privacy in vehicular communications by extending the simulation

by own detectors. Morenz (2008) built a system for predicting public transports' travel times by adapting the simulation state to measured induction loop values online.

As a summary, it should be noted that SUMO has proven to be extensible, not only by its original developers but also by external contributors. This was one of the initial goals which have been achieved. Also, SUMO's capability to simulate large networks has proven to be a great feature for investigation of large-scale effects of new methods or technologies.

7.2 Model Building Principles in SUMO

Because SUMO is not an established software suite, used networks and demands are usually given in the formats used by other simulation packages. Due to this, much work within the development of SUMO had been spent on implementing methods for importing road networks and demand data. What was a need at first got a philosophy over time: the main idea when preparing road networks or demands for a simulation is to have them be generated from digital descriptions and enrich or process them for being usable as input data to the simulation. This is one of the reasons that SUMO does not have a network editor or an integrated editor for the demand, yet (but see Section 7.7). Both preparing the road network and the demands are described in detail in the next sections, followed by a summary on preparing simulation scenarios. Notes on validating a scenario are given in Section 7.5.

7.2.1 Preparing a Road Network to Simulate

When dealing with the simulation of real-life road networks with SUMO, the common approach is to import an available digital road network. SUMO's network importer, a tool called "NETCONVERT," is able to read networks from VISUM, TIGER, ArcView shape files, Vissim, Robocup Rescue League folders, OpenStreetMap, and a "native" XML representation of the road network graph. As one may note, most of the named road networks are not originally designed for being used for a microscopic simulation of traffic. That's why information important for a correct microscopic traffic flow simulation, such as which lane may be used to get to a following road, right-of-way rules on intersections, and even information about traffic light positions or their programs, is often missing.

NETCONVERT tries to help here, by applying heuristics for computing missing values. Some of the heuristics are applied only if certain information is not given. As an example, connections between edges will be determined heuristically for ArcView shape files, which do not contain them. NETCONVERT's heuristics include the following:

- Computation of the turning direction for each road (what is also necessary for later computation of connections between lanes).
- Computation of connections between lanes.
- Computation of intersection types (right-before-left vs. intersections with a high-priority road).
- Computation of right-of-way rules for the roads participating in an intersection.
- Computation of road and intersection geometries.

Additionally, NETCONVERT includes heuristics for determining positions of traffic lights, their programs, and for determining where additional lanes for highway on- and off-ramps should be added.

All in all, the implemented heuristics are a great help when dealing with networks which do not contain the needed information. Usually, one obtains a network which looks realistic with a single call to NETCONVERT. Nonetheless, further, manual inspection and work on the networks are usually needed. This means that in practice, the following procedure must be performed in order to prepare a network for simulation:

- An available digital road network description is imported and written into SUMO-native XML files that describe the road network in terms of nodes (intersections/junctions), edges (roads), and lane-wise connections between them.
- The XML files are converted into a road network that may be used within the simulation.
- The resulting road network is inspected.
- In the case the resulting road network is erroneous, these errors are corrected within the XML files that were used to build it.
- The process iterates until the network topology has a proper quality.
- After the network's topology is fixed, traffic light information must be mapped onto it, mainly by replacing the ones NETCONVERT computes by definitions that are based on signal plans from the real world.

It may be interesting whether all imported formats are used similarly. This should be strongly denied. Within our projects, mainly networks imported from VISUM and from NAVTEQ are used; the second either encoded in a proprietary DLR-internal format or as ArcView shape files. From time to time, the definitions of the road networks to simulate reach us in the form of Vissim simulations. In the past time, we have also put an incremented effort in importing networks from the Open Street Map (OSM, 2009) project in order to obtain real-world examples we are legally able to make public. All other import facilities were mostly implemented for evaluating them for their usability but were not thoroughly tested. Of course, this affects the qualities of the import functions. Formats which were only briefly investigated may contain needful information which is not imported. Also, changes in the format and problems one may encounter if using a certain format get only visible if the format is used frequently. Though no explicit issues are known,

it should be mentioned, that the importer for TIGER networks, which are very popular within the vehicle-to-vehicle communication community, is not supported well.

7.2.2 Preparing the Demand

Choosing a method for demand modeling depends on the research topic under consideration, the study area, and the availability of data about this area's traffic. Because SUMO was designed for the simulation of a synthetic population which consists of single persons with distinct routes and explicit departure times, SUMO's "native" demand definition is a list of vehicle departure times with the roads the according trip shall start and end at. Using these definitions, the according complete path through the network can be computed, using a simple shortest path algorithm, yielding in a list of vehicle routes. Such routes through the network – together with their start time and possible additional information about the vehicle – are the input required by the simulation. Additional methods for computing the dynamic user equilibrium by a simulation-based dynamic traffic assignment are available and yield a usable, realistic, and fine grained demand and load description of the study area.

Of course, such "microscopic" demand descriptions are not available at all. Only systems which are based on synthetic populations or agents are able to deliver per-vehicle information about the departure time and this vehicle's trip origination and destination road. Though such systems exist, defining a demand for a certain area is very time consuming, because many sociological data must be gathered and used. This means, that even in our own projects, we always had to "fall back" to common and established descriptions of road traffic demand. For large-scale scenarios covering complete cities, importing origin-destination (O/D) matrices from VISUM or other formats has proven to yield valid results. The import consists of the following two steps, which are both directly supported by tools of the SUMO suite:

- Convert the O/D matrix or matrices into a list of single-vehicle "trip" information, consisting of the departure time and the origin and the destination road.
- Perform a dynamic user assignment in order to obtain a realistic set of routes through the road network.

Such demand descriptions should of course consist of a set of O/D matrices, each defining only a small time window, preferably 1 h or less. O/D matrices which describe a whole day's traffic aggregated into a single O/D matrix are rather too coarse for their direct application in a microscopic traffic simulation. For matrices covering a whole day, SUMO itself allows only to apply a user-defined time line of the given amount's percentages over a day. Processing of multiple O/D matrices is supported directly.

It must be noted that O/D matrices may differ very much in their granularity and how connections between districts and the underlying road network are set up. Though it is possible to map a district's area to the edges it contains and use these as trips' origins and destinations, a valid methodology for this purpose was not yet evaluated or even implemented. The consequence is to use the given connections between districts and the road network. This means in practice that a valuable O/D description should use minor roads of the network as connection to the districts. The contrary – additional connections, which have no counterpart roads in the real world's road network, pointing to major intersections, maybe even ones controlled by traffic lights – should be avoided in any microscopic traffic simulation, because these intersections' attributes and the behavior of vehicles around them diverges strongly from reality.

Fine grained, time-dependent O/D matrices are not always available. Also, in the case of smaller areas, available O/D matrices may be too coarse, because the needed traffic description is wanted to be given on a per-road base, and not by joining roads into districts. For fulfilling this need, the SUMO suite includes two further applications which generate per-vehicle demands. The first one is a router based on turning ratios at intersections. This tool reads definitions of flows and time-dependent turning ratios at intersections. Each flow is described by its starting edge and its amount. Using these data, the tool computes the per-vehicle demand, again as a list of vehicles with their routes and departure times by simply choosing a certain continuation at each junction a vehicle passes according to the given ratios. It should be noted that the procedure is very simple and straightforward, so that its usage is easily set up and the results should be valuable because being easy to evaluate. Note, however, that by using this tool in larger networks is expected to yield weird results, since no guard is build in to hinder the routes from forming loops. Note in addition that this tool is not frequently used at our institution, because collecting turning ratios is only possible for smaller areas, while our work concentrates on large networks. It is not known whether or how often this tool is used by other users.

The second tool uses time series of traffic flow from observation locations. At first, these are classified, marking observation points which have no observation point in front as "sources," and those having no observation point behind as "sinks." Then, streams are inserted at the "source" positions and propagated through the network until they reach a "sink." At each junction at which more than one continuation is possible, the streams are spread in accordance to the following observation points' measures.

This principle is very simple and has the disadvantage that only the streams' distributions across the network are computed, not the vehicles' real routes. Nonetheless, it has proven to be valuable for populating a simulation scenario if only the amounts of vehicles passing observation points must be investigated. The major problem in its usage originates from the need to have all entries and exits of the modeled network to be covered with measures. This is usually not the case. Also, this approach works only for networks where no loops in vehicle routes can occur, making it inappropriate for larger inner-city regions but very well suited for highway corridors.

7.2.3 Summary on Preparing a Simulation Scenario

Following the decision to built road networks fast by importing them, no graphical editor which could support a user by adapting changes manually is available by now. This in fact makes the work for getting a complete road network for large areas very uncomfortable and one should state that the original idea to have the networks imported fast is contradicted partially here by.

The tools for demand generation which are included in the package can deal with standard demand descriptions used within the traffic research and offer further possibilities to generate traffic when real-world data are used. Nonetheless, an additional effort is needed for converting given data for using it as input for these tools.

7.3 Fundamental Core Models

As common to most microscopic simulations, the models for vehicles' longitudinal (car-following) and lateral (lane-changing) behavior are executed separately within SUMO. They interact in a minor manner, as described within the subchapter on lane changing, but a close coupling has not yet been done. At first, the vehicles' lateral movement is computed, their lane-changing in a second step. In the following, both models are described in this order, followed by an outlook on further research.

7.3.1 Longitudinal Vehicle Movement

SUMO uses a modified version of the time-discrete and space-continuous car-following model by Krauß (1998). The model is based on a derivation of a safe gap a vehicle, the EGO, needs to stop behind a leading vehicle, the LEADER, without colliding with him. Both the vehicles' maximum decelerations (assumed to be equal) and EGO's reaction time are considered. By using the usual approximation for the braking distance d(v) = v^2/(2b), the following formula for determining a safe speed can be computed. This safe speed, given the distance to the LEADER and the speed of the LEADER, assures a collision-free behavior:

$$v_{\text{safe}}(t) = -\tau \cdot b + \sqrt{(\tau \cdot b)^2 + v_{\text{leader}}(t-1)^2 + 2 \cdot b \cdot g_{\text{leader}}(t-1)}$$

where

$v_{\text{safe}}(t)$ is the safe velocity for time t (in m/s);
τ is EGO's reaction time (in s);
b is the maximum deceleration ability (in m/s^2);

$v_{\text{leader}}(t)$ is LEADER's velocity at time t (in m/s);
$g_{\text{leader}}(t)$ is the gap (between EGO's front and LEADER's back) at time t (in m).

Now, it must be assured that EGO neither accelerates faster than it is able to do nor that it gets faster than its maximum velocity. The result is the "desired" or "wished" velocity, computed as

$$v_{\text{des}}(t) = \min\{v_{\text{safe}}(t), v(t-1) + a, v_{\max}\}$$

where

$v_{\text{des}}(t)$ is the velocity EGO wants to use (in m/s);
$v(t)$ is EGO's current velocity (in m/s);
a is EGO's maximum acceleration ability (in m/s^2);
$v_{\max}$ is EGO's maximum velocity (in m/s).

One major achievement of Krauß' model is to assume a driver is not perfect in realizing the desired speed. Instead, the speed actually chosen is somewhat smaller, and this adds important features to the behavior of the model. For example, this random difference to the desired (optimum) speed leads to spontaneous jams and the slow-to-start characteristic of real drivers. The model implements this driver imperfection by a stochastic deceleration. For obtaining the EGO's next speed, it has to be assured that the vehicle is not moving backward afterward. This makes the last step of computing the vehicle's speed be

$$v(t) = \max\{0, v_{\text{des}}(t) - r \cdot a \cdot \varepsilon\}$$

where

r is a random number, between 0 and 1;
ε is EGO's driver imperfection, between 0 and 1;
$v(t)$ is EGO's final speed for time t (in m/s).

Two extensions have been added to the original model. The first was to apply a decay into the ability to accelerate with increasing speed. As a simplification, this was modeled using a linear function, yielding a speed-dependant acceleration computed as following:

$$a(v) = a\left(1 - \frac{v}{v_{\max}}\right)$$

The second was to reduce the driver imperfection's effects on accelerating from low velocities. Instead of the prior computation of dawdling, a distinction based on the vehicle's speed is done, so that the vehicle's final speed is computed using

$$v(t) = \max\left\{0, v_{\text{dawdle,new}}(t)\right\}$$

where

$$v_{\text{dawdle,new}}(t) = \begin{cases} v_{\text{des}}(t) \cdot \varepsilon \cdot r & \text{if } v_{\text{des}}(t) < a(v_{\text{des}}(t)) \\ v_{\text{veh}}(t) - \varepsilon \cdot r \cdot a(v_{\text{des}}(t)) & \text{otherwise} \end{cases}$$

The model by Stefan Krauß is very fast in execution due to the small number of needed computations and has proved to be valid enough in comparison with other models (see Brockfeld et al., 2004; Brockfeld and Kühne and Wagner, 2004; Brockfeld and Kühne and Wagner, 2005 and Section 7.5). The used driver imperfection has found acceptance as an extension to famous models such as the Wiedemann model (see Brilon et al., 2005).

Nonetheless, having a car-following model is only one part of modeling the longitudinal behavior of a driver-vehicle instance. The original model's applicability for traffic simulation was demonstrated by simulating traffic on a large circular road with the parameters of highway traffic. It is not surprising that when moving to complex scenarios, which include networks with roads of arbitrary lengths, complex right-of-way rules and traffic lights, and different vehicle routes, further methods had to be implemented for making the simulation work at all.

First of all, each vehicle must take into account the infrastructure in front of it. Changes in the speed allowed on the approached roads must be regarded before entering the road. Looking ahead must also be applied in order to follow the right-of-way rules without letting the vehicles decelerate stronger than they are declared to be able to. For assuring a collision-free behavior in networks with a complex infrastructure, the following computations are done:

- Adapt velocity in dependence to LEADER's speed and the distance to him (over the next lanes) as described in the original Krauß model.
- Adapt the speed allowed on the next lane.
- If EGO has no right-of-way on the next intersection, compute two velocities, one for passing the intersection and one based on the assumption that the vehicle has to brake and stop in front of the intersection. Store them. Let the intersection know the vehicle is approaching.
- Continue with the next lane along the route or stop, if the seen lane lengths' sum is larger than the braking distance.

After these steps have been performed for all vehicles, it is decided which vehicles are allowed to move over the intersections by following the intersections' right-of-way rules. The vehicles are moved in accordance to their so computed rights-of-way afterward using the previously stored velocities.

As mentioned, the Krauß model has proven to be valid, usable, and fast. Nonetheless, some issues were noted and should be mentioned:

- The driver imperfection is modeled in a very simple way. It has been not validated against real trajectories and should be assumed to resemble traffic flow's macroscopic behavior, not the (microscopic) behavior of a single driver.
- The simplification done by using a reaction time equal to the model time step of one second is known to be problematic within dense scenarios. Here, often the real flow cannot be reproduced.

7.3.2 Lane-Changing Model

While the longitudinal model has proven to be robust so that no major changes were needed, the lane-changing model is strongly evolving since SUMO's begin. The reason is that the original model formulated by Krauß only incorporates a driver's tactical decisions, mainly formulating a driver's behavior based on the assumption a driver wants to drive fast. The navigational (or strategic) part of choosing a lane – the need to change to a certain lane in order to be able to continue the route – is not regarded.

The currently implemented lane-changing behavior (see also Krajzewicz, 2009, in German) solves the problem by computing a valid path through the network, in the means that the chosen lanes can be used for continuing the route; from now on the term *valid lane* will denote a lane which may be used for continuing the route without the need to change the lane. Each lane of the road EGO is currently at and of the roads following along its route – up to a viewing distance – is examined. Besides the distance EGO may continue using the regarded lane without the need to change to a valid lane, the lanes' occupancies are collected. Given these descriptions of lanes, it is decided for EGO whether a lane change into the direction of a valid lane is needed. This is the case if EGO's distance left to the position from which the route cannot be continued is lower than an assumed distance needed for the lane change. The assumed needed distance is computed using

$$d_{lc,\mathrm{veh}}(t) = \begin{cases} v_{\mathrm{veh}}(t) \cdot \alpha_1 + 2l_{\mathrm{veh}} & \text{if } v_{\mathrm{veh}}(t) \leq v_{\mathrm{swell}} \\ v_{\mathrm{veh}}(t) \cdot \alpha_2 + 2l_{\mathrm{veh}} & \text{otherwise} \end{cases}$$

where

$d_{lc,\mathrm{veh}}(t)$ is the assumed distance vehicle *veh* needs for a lane change in time t (in m);
$v_{\mathrm{veh}}(t)$ is vehicle's *veh* speed at time t (in m/s);
v_{thresh} is a threshold discriminating highway and urban behavior (in m/s, set to 14 m/s);
α_1,α_2 are scaling factors (currently set to 5 and 15 s, respectively);
l_{veh} is vehicle's *veh* length (in m).

The approach takes into account the occupancies of the lanes that must be used until reaching the position where the route cannot be continued, including the

current lane, the target lane, and lanes to pass. Therefore, the lengths of vehicles in front of the regarded vehicle are subtracted from the distance left. This forces the simulated vehicles to change the lane at the end of a queue on the destination lane, preventing them from trying to drive beside a jammed lane, first, and then trying to merge into this jammed lane when no further continuation is possible.

For the opposite direction – moving away from a valid lane – similar tests are done. EGO is only allowed to move into the direction if this lane change and the lane changes needed to come back to a usable lane are possible within the distance left, regarding the lanes' occupancies.

For the tactical part of the lane changing – the wish to move forward fast – an approach based on Ehmans (2001) was chosen. During his drive, a driver stores the benefits of changing the lane. The benefit to change a lane is the difference between the safe speeds on the neighbor and on the current lane, computed using the car-following model, and normalized by the maximum velocity the vehicle could use under free-flow condition:

$$b_{l_n}(t) = \frac{(v_{\text{poss}}(t, l_n) - v_{\text{poss}}(t, l_c))}{v_{\max}(l_c)}$$

where

$b_{l_n}(t)$ is the benefit of a vehicle to change to lane l_n at time t;
l_c and l_n are the vehicle's current and neighbor lanes, respectively;
$v_{\text{pos}}(t, l)$ is the velocity the vehicle could drive safe with on lane l at time t (in m/s);
$v_{\max}(l)$ is the maximum velocity the vehicle can take on lane l (free flow, in m/s).

Using the benefits for neighbor lanes, a driver-internal "memory" variable, which represents the simulated driver's wish to change to a neighbor lane, is adapted. If the benefit to change the lane is >0, this benefit is added to this memory, signed by the direction. If the benefit is <0, the current lane is faster than the neighbor lane, the memory value is divided by two, suppressing the wish to change into this direction.

A lane change is initiated if the absolute value of the memory variable is larger than a certain threshold. The sign of the memory variable represents the direction of the lane change. Of course, a lane change is only possible if the lane EGO wants to change to has enough space at EGO's current position. Additionally, the resulting gaps must allow further collision-free continuation of driving.

In the case the situation does not allow EGO to enter the desired lane, he starts to interact with the vehicles which are in front and behind him at the this lane. The vehicle itself and the vehicles at his destination lanes are adapting their speed in dependence of whether they are blocked/blocking at their front or their back using the following rules:

$$v_{\text{next}}(t) = \begin{cases} v_{\text{decel}}(t) \text{ if blocking/blocked at own back and front} \\ v_{\text{decel}}(t) \text{ if blocking/blocked at own front} \\ v_{\text{accel}}(t) \text{ if blocking/blocked at own back} \end{cases}$$

where

$v_{\text{accel}}(t) = v_{cf}(t) + \frac{v_{\max}(t)}{2}$ is the vehicle speed after accelerating (in m/s);
$v_{\text{decel}}(t) = v_{cf}(t) + \frac{v_{\min}(t)}{2}$ is the vehicle speed after decelerating (in m/s);
and
$v_{cf}(t)$ is the car-following speed (including the driver imperfection, in m/s);
$v_{\max}(t)$ is the speed after a maximum possible (in accordance to car-following) acceleration (in m/s);
$v_{\min}(t)$ is the speed after a maximum possible deceleration (in m/s).

The model behaves well for both highway and urban scenarios, assuring the vehicles are choosing their lanes early enough and also assuring that all available lanes are used. Nonetheless, the realized look-ahead along the roads to pass does not consider the behavior of other vehicles. This is problematic as soon as a vehicle's current lane is blocked by standing vehicles, but must be soon used for continuing the route. In these cases, the model tends to suppress the vehicle to change the lane.

7.3.3 Summary on Used Models

The initially implemented models for car-following and lane-changing have evolved by incorporating methods for taking into regard the road infrastructure in front, including the right-of-way rules, and the occupation by other vehicles. They are applicable and valid for most cases. Nonetheless, unwanted behavior was observed for both. To solve these issues, and for allowing further applications, the development will continue. Though its main focus is to allow simulations with time steps <1 s currently, also further work on lane-changing is meant to be done.

7.4 Dynamic Traffic Assignment

The ability to compute a dynamic traffic assignment is an integral need for proper simulation of large area scenarios. SUMO uses an approach developed by Gawron, 1998 where each vehicle (a) has its own route, (b) knows its own travel time through the network, and (c) computes new routes without taking into account the travel times of other vehicles. This algorithm is driven by the travel times the used simulation model computes and not based on assumptions how real drivers choose a route through the network. It converges toward an equilibrium. It is an iterative approach, working as following:

1. Initialize the process by computing the fastest route through the empty network for each vehicle to simulate and add this route to the list of routes known by this vehicle. The probability to use this route by this vehicle is set to 1.
2. Perform the simulation using the currently chosen routes in order to obtain the edges' travel times over simulation time.
3. Compare the mean travel times against those obtained in the previous run (if any) and quit if the algorithm converges, i.e., the mean travel time reduction falls below a given threshold.
4. Compute new routes for all vehicles using the network's current travel times. If a new route for a vehicle was found, add it to the list of routes known by the vehicle. Update all known routes' estimated travel times and their probabilities to be chosen. After that, choose one route for this vehicle taking into account the route choice probabilities and continue with step 2.

In the following, the methods for adapting the route probabilities to the last iteration's travel times are shown. At first, the travel times for the routes known by a driver are adapted to the travel times obtained from the simulation using the following formula:

$$\tau'_d(x) = \begin{cases} \tau_s(x) & x = \text{last chosen route} \\ \beta\tau_r(x) + (1-\beta)\tau_d(x) & \text{otherwise} \end{cases}$$

where

$\tau_d(x)$, $\tau_s(x)$, $\tau_r(x)$ are route x's prior travel times as estimated by driver d, retrieved from the simulation, and reconstructed from the edge travel times that were determined by the simulation, respectively (in s);

$\tau'_d(x)$ is driver d's new estimation of the duration of route x (in s);

β is a factor affecting the speed of adapting remembered travel times to the current.

Using these adapted travel time information, the probabilities to choose one of the known routes are recomputed. The probability for each unused route known by the driver is recomputed by a function that compares its travel time with the travel time of the route used in the last simulation step using the following formula:

$$p'_d(r) = \frac{p_d(r)\ \ (p_d(r) + p_d(s))\ \exp\left(\frac{\alpha\delta_{rs}}{1-\delta_{rs}^2}\right)}{p_d(r)\exp\left(\frac{\alpha\delta_{rs}}{1-\delta_{rs}^2}\right) + p_d(s)}$$

where

$p_d(x)$ is the prior probability to use route x by driver d;
$p'_d(x)$ is the new probability to use route x by driver d;
r is the route used in the last simulation run;
s is another route from the list of known routes;
δ_{rs} is the relative cost differences between routes r and s, computed as

$$\delta_{rs} = \frac{\tau_d(s) - \tau_d(r)}{\tau_d(s) + \tau_d(r)}$$

where

$\tau_d(x)$ is the travel time for driver d to complete route x.

The probability to use the route which was already used in the last iteration step is updated by

$$p'_d(s) = p_d(r) + p_d(s) - p'_d(r)$$

Normally, travel times for edges are collected and aggregated into intervals of 15 min during the simulation's runs. The so obtained time series of edge travel times are then read by the router module and used for the described computation of new routes and probabilities to use known routes. During this process, each edge's travel time for a vehicle's entry time is determined by looking up in the corresponding time series for the interval that matches interval begin $\leq$ entry time $<$ interval end. For α and β, usually values of 0.5 and 0.9 are used, respectively.

The algorithm has proven to generate valuable results. Nonetheless, its iterative nature makes it very slow in execution – in order to get usable assignments, often more than 20 iterations are necessary, each consisting of a computation of new routes and a simulation step. In addition, as vehicles are starting using the fastest routes in an empty network, without an a priori assignment, the first iterations are dominated by large jams, making the simulation additionally slower than the normal execution time.

To solve these problems, several attempts have been undertaken, including implementation and evaluation of macroscopic traffic assignments, and introducing methods which try to solve the problems of slowing down the simulation by jams during the first simulation steps. A report on these methods can be found in Behrisch et al. (2008) and Behrisch et al. (2008). The most promising – and surprising – attempt is the usage of a one-shot assignment. Here, each vehicle is started with its origin and destination edge, the route is then computed at the time the vehicle enters the network. The network's edge weights (travel times) used for computing the currently fastest route are adapted to the situation within the network in each time step using the following formula:

$$w(t,e) = \begin{cases} l(e)/v_{\max}(e) & t = 0 \\ w(t-1,e) \cdot r + l(e)/v_{\text{curr}}(t,e) \cdot (1-r) & \text{otherwise} \end{cases}$$

where

$w(t,e)$ is the weight of edge e at the current simulation step t (in s);
$l(e)$ is the length of edge e (in m);
$v_{\max}(e)$ is the maximum velocity allowed on edge e (in m/s);
$v_{\text{curr}}(t,e)$ is the mean velocity of vehicles on edge e in time step t (in m/s);
r is a remembering factor.

As mentioned, the results of using this approach were surprisingly good, combining a fast execution with short travel times – the measure used for determination whether a network equilibrium is approached – of computed routes. Though, further investigations are still necessary.

7.5 Calibration and Validation

Within the description of the used longitudinal model the need to distinguish between the used car-following model and what the complete simulation does was already mentioned. Because a traffic simulation is a computer application, the validation must be done at different levels, starting at a verification of the computer program as such. This is worth to mention because one can learn from different possibilities to verify a computer application about meaningful verification of models. The current attempt to assure SUMO's correct behavior assumes the following levels:

- Unit Tests
  Unit Tests are very small tests. Each assures that a certain function – the minimal part of a computer application – behaves as should given a set of parameters or given a certain internal state. An example would be to test whether a multiplication function really returns 4 if two parameters, 2 and 2, are given. Of course, most of an application's functions are more complicated. Currently, the usage of unit tests within SUMO is being evaluated, and only a few tests are available by now, written using the googletest framework (2009).
- Acceptance Tests
  An acceptance test compares the output – including what is printed on the command line and the generated files – of a complete application's execution against what was declared to be correct. If the current and the last outputs are same, the test returns a message about a correct behavior of the software, otherwise it reports an error. The SUMO suite is tested each night using more than 2000 acceptance tests, of which almost 800 deal with the simulation itself. The test suite is set up using the "TextTest" (2009) framework.

  The major lesson learned during the work on the tests was to make each test as simple as possible. The verification of large tests, tests including interactions between many vehicles, for example, is time consuming and error-prone. Also, complicated tests are also more sensitive to small changes of the model. The reason is that the generated files are directly compared against each other and a minor change in the output, for example, a difference of a vehicle's speed by 0.01 m/s already sets the test to have failed. This requires the cumbersome verification process to be redone. This is rather not the case if the tests have already been initially set up well defined.

  As a conclusion, it should be stated that acceptance tests are not a proper tool for assuring a simulation's correctness for complex scenarios. Still, they are very valuable for assuring the correct working of an application.

- Model Tests
  Beside the comparison to the fundamental diagram given in Krauß (1998), the car-following model by Krauß was tested within a set of model comparisons together with other microscopic traffic flow models. The reported results (Brockfeld and Wagner, 2004; Brockfeld et al., 2004; Brockfeld et al., 2005) show its applicability to represent real traffic flow and real driver behavior.

The presented possibilities to test a simulation are not yet covering all functions the suite's applications offer. Of course, this will never happen at all, since a complete coverage of possible settings would not be possible due to their infinite number. This means, that a verification of a setup simulation scenario is still necessary. The validation procedure performed within our projects depends mainly on the available data. Normally, induction loop values are available and the simulation is compared against these. This is done by inserting simulated induction loops at the position of the induction loops in the real world. The values generated by these after a simulation's run are directly – despite the normally needed formatting conversion – comparable with real induction loop values. The results are assumed to be valuable because they show whether the modeled flow is correct and is correctly propagated through the road network. In addition, it may be evaluated whether the simulated vehicles' speed matches the reality at the positions the induction loop are placed.

7.6 Extended Modeling Capabilities: Working with External Applications

TraCI, the "*Tra*ffic *C*ontrol *I*nterface," is the contribution done by an external institution of which SUMO benefits most. TraCI extends SUMO by the possibility to interact with a running simulation online by connecting an external application to SUMO using sockets. It was implemented by staff members of the University of Lübeck, mainly Axel Wegener (see Wegener et al., 2008; Wegener 2008). When used, the simulation is triggered from the external application to continue with the next step. This means that in contrary to a "normal" simulation, each step must be explicitly called by the external application. As a result, both the simulation and the external application run synchronously.

TraCI allows asking for attributes of vehicles, traffic lights, induction loops, road infrastructure, and other simulation objects. Using TraCI one can also influence simulated objects. The phase of a traffic light, its duration, or even a complete program of a traffic light can be changed using this interface. It is also possible to control a traffic light completely via this interface, setting explicit states for all signals. TraCI also allows changing a vehicle's maximum speed, forcing it to brake, or to change the lane, give it a new destination or to force a recomputation of a vehicle's route.

The major application of TraCI is to connect SUMO to the communication network simulator ns2 – either directly or via an application in between, such as

TraNS (Piórkowski et al., 2008; EPFL, 2008), which is developed mainly by Michał Piórkowski and Maxim Raya from the EPFL Lausanne. TraNS allows defining V2V applications for their simulation using ns2 and SUMO and is very prominent within the V2V community. The support for TraCI is enabled within SUMO by default since version 0.9.8. Due to being used within the currently running project iTETRIS (see Section 7.7), TraCI's capabilities are ongoing a process of revision and extension.

7.7 Selected Projects, Contribution, and Data

Two projects the DLR is participating in are worth being elaborated more deeply. The first one, DELPHI, is a portal accessible to authorities for managing own reaction forces and the road traffic itself in the case of catastrophe or large event scenarios. The second one is iTETRIS, a project founded by the European Commission which is aimed at establishing a common platform for development and evaluation of traffic management strategies based on V2V/V2I communication. These projects are described in the first two of the following subchapters. Then, a foreign application meant for being used in conjunction with the SUMO package, the "SUMO Traffic Modeler," is described. At last, a large scenario named "TAPAS Cologne" that was made public in the recent time (end of 2008) is introduced.

7.7.1 DELPHI

DELPHI – Deutschlandweite Echtzeit Verkehrs-Lage und Prognose – is the continuation of two former projects, Weltjugendtag 2005 (WJT2005) and Soccer (2006) which gained large public interest. The major scientific challenge was to gather information about the current traffic situation using airborne surveillance systems and to embed these in a simulation-driven representation of the road's traffic state together with conventional induction loop data. The so obtained representation of the real-world traffic was extrapolated half an hour into the future.

DELPHI continues this approach, aiming for (a) a sustainable delivery of the road network's current and future state and (b) offering the authorities to manage their law enforcement and emergency services using this data. Two major German cities are currently covered by the system, Cologne and Munich. DELPHI is web based; it retrieves induction loop values from the local highway administration offices via a dedicated connection. In addition, measures from airborne detectors, developed within the DLR project "ARGOS" and floating car data (Schäfer et al., 2002) are received and included, if available. The system is accessible for a user using an internet browser application.

Besides being shown directly to the user, the traffic information gained from sensors is integrated into a simulation. The simulation itself was targeted at the traffic situation on average weekdays and weekends. This was done by importing networks from NAVTEQ and demands from the VALIDATE data set by PTV AG,

first. In further steps, the given demand was assigned to the network, and afterward calibrated to stored induction loop measures using a matrix adaptation approach.

During the system's operation, the simulation is re-started every 5 min, starting to simulate the time 5 min before its execution time. Besides loading the precomputed demand for the time to simulate and the last simulation state, the last induction loop and airborne detector data are used for calibrating the flows and their speeds within the simulation. For the simulation's first 5 min, the collected data is used. The so obtained network state – calibrated to measured values – is stored for the next simulation run. Further 30 min of traffic are then simulated, being additionally calibrated by extrapolated values. Besides extrapolating the state into the future, the simulation also models traffic on roads which are not covered by detectors this way.

This simulation speed – more than six times real time for study areas as large as Munich and Cologne which both have more than one million inhabitants – can no longer be achieved by a pure microscopic simulation. For realizing the system, SUMO was extended by a mesoscopic queue model, originally developed by Gawron (1998), and extended and made more realistic by Eissfeldt (2004). The model has been embedded into SUMO with no change on the interfaces; in order to enable it, SUMO has to be started with only one additional parameter. This allows reusing all available applications from the SUMO package with no change. Note, however, that the mesoscopic extension is not available as a part of the open-source package.

The obtained traffic representation is used by the DELPHI system to allow the user to (a) compute shortest routes, regarding the current traffic state, (b) monitor these routes, and (c) compute isochrones of accessibility. This is enhanced by functionality specifically requested by the authorities which will result in a full-fledged web-based (traffic and event) management tool as the project continues.

7.7.2 iTETRIS

As noted before, SUMO is used often by the communication network simulation community in the context of vehicle-to-vehicle and vehicle-to-infrastructure communication. iTETRIS – an Integrated Wireless and Traffic Platform for Real-Time Traffic Management Solutions – is a project founded by the European Commission aiming at this topic, still with a clear focus on traffic management. The work done in iTETRIS is meant to "... create a long term (beyond the project), global, sustainable, open, vehicular communication and traffic simulation platform facilitating large-scale, accurate, multidimensional evaluation of cooperative ICT solutions for mobility management ..." (iTETRIS, 2009). This goal shall be achieved by work on the simulators themselves – ns3 for networking simulation and SUMO for road traffic simulation – and on the connection between them.

The project builds upon a real city's traffic problems; the situation in the city of Bologna, which administration is one of the project's consortium members, is described and evaluated showing bottlenecks and proposing solutions for solving

these. Traffic descriptions for this city's problematic areas are supplied which are then translated into SUMO format from the original Vissim and VISUM sources. Given these evaluations and scenarios, traffic management strategies, which are assumed to be capable to solve the reported problems and which are based on vehicular communication, are derived, implemented, and tested.

A major topic within this project was to assure that the environmental effects of the evaluated strategies will be considered. Therefore, the possibilities to compute gas pollutant (CO, CO_2, NOx, PMx, HC) emissions, fuel consumption, and noise emissions were implemented in SUMO. Gas pollutants emission and fuel consumption were modeled based on HBEFA (INFRAS 2009), a database on vehicular emissions. The values of the HBEFA database were approximated by functions, first. Then, the resulting function parameters were clustered in order to obtain a set of vehicle classes which is smaller than the original one which consists of over 100 classes, in order to ease setting up a simulation scenario. For the noise emission model, Harmonoise (Nota and Barelds and van Maercke and van Leeuwen, 2005) was chosen and implemented. The implementations of both models in SUMO allow computing emissions on per-vehicle, per-lane, and per-road base. The two latter also allow different time aggregation of the values. The implementation of both models is completed and available as an integral part of SUMO since March 2009 (release 0.10.2).

A second major extension of SUMO for its usage within iTETRIS is to allow SUMO to run with time steps smaller than 1 s. This will be achieved by implementing a recently developed car-following model. In addition, the rules for regarding the right-of-way on junctions must be reformulated, because they are currently coupled tightly to the simulation's time step length. Further topics of this project are aiming at extensions toward further possibilities to interact with external applications via TraCI to allow them (a) to control the currently simulated traffic lights, (b) reroute currently simulated vehicles, and (c) allow the simulation of advanced driver assistance systems ADAS based on vehicular communication.

7.7.3 SUMO Traffic Modeler

"SUMO Traffic Modeler" (Papaleontiou and Dikaiakos, 2009) is a graphical editor for traffic demands for a given network. It was written by Leontios Papaleontiou as a part of his Master thesis done at the University of Cyprus. SUMO Traffic Modeler allows loading an existing SUMO-network and to edit "traffic area elements," similar to districts in user assignment tools, graphically. The shape of a traffic area element may be either a polygon or a circle.

Besides defining traffic area elements as such, SUMO Traffic Modeler allows to model demands between them, or use one as a "hot spot" – an area within the road network where vehicles preferably start or end. A further area element type allows to model activity-based demand generation, using a simplified synthetic population approach. Furthermore, demands can be also edited by giving an origin and

a destination road. The so generated demands can be exported as per-vehicle trip definitions which can be further processed by tools from the SUMO suite.

The Traffic Modeler is a very interesting application, since it adds a traffic demand modeling tool to SUMO which was missing before. Additionally, it is the first tool which was implemented at an external institution and which is mainly related to questions from traffic modeling and simulation.

7.7.4 TAPAS Cologne

Making the "TAPAS Cologne" scenario available is an approach to supply a high-quality example data set which includes all data needed for performing a simulation and which can be used as a base for own evaluations. The amazing OpenStreetMap (2009) project delivers a free digital road network, but traffic demands are normally not freely available. Data from the TAPAS project (Hertkorn and Kracht, 2002; Hertkorn and Wagner, 2004; Hertkorn), which was already mentioned in Section 7.1, could have been made freely available earlier, because they were generated at the DLR, but the originally used digital road network could not be put in the public domain. After projecting the demand data onto an OSM network, we can now offer a large area covered completely by a normal day's passenger traffic. Applying OSM license, the data is available under the "Creative Commons Attribution-Noncommercial-Share Alike 3.0" license (2009), what means that the data must not be used for commercial purposes, and we have to be notified in the case someone uses it.

The data set in its current form contains (a) the road network imported from the OSM database with no changes, despite applying heuristics for building highway on- and off-ramps and traffic light programs; (b) a set of points of interest and polygons extracted from the OSM database which represent buildings such as shops, hospitals, etc., parks, parking places, and positions one can find public telephones or one can give his bottles back; and (c) the demand of passenger cars for the given road network in form of routes through the network. In addition, the configuration files for the SUMO simulation are given, so that one can execute the scenario out of the box (Fig. 7.1).

The currently given data have a large potential. In combination with the given – and increasingly growing – information about the area in the means of points of interest and polygons, this data set allows to simulate location-based services, planning routes for emergency vehicles, and much more. The availability of information about railways and public transport stops should allow multi-modal simulations.

Note, however, that still a lot of additional work is needed to make this data set really usable. The positions and programs of traffic lights must be revalidated. The same holds for the roads' numbers of lanes and the intersections' right-of-way rules. Public transport lines must be set up as well as delivery traffic, which is not a part of the TAPAS data set.

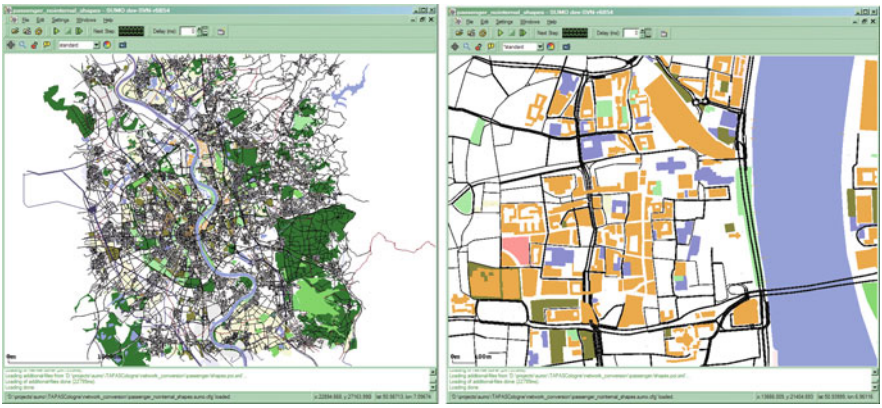


Fig. 7.1 An overview (*left*) and a detail view (*right*) on the TAPASCologne scenario

We hope, that the scenario will be incrementally improved, starting with solutions which are probably working, but not yet completely based on values from the real world. Nonetheless, this work is not meant to be done by DLR only. We hope on the contribution from external participants.

References

Krajzewicz D, Hertkorn G, Rössel C, Wagner P (2002) SUMO (Simulation of Urban MObility) – an open-source traffic simulation. In: Al-Akaidi A (ed) Proceedings of the 4th Middle East symposium on simulation and modelling (MESM20002), S. 183–187, 4th Middle East Symposium on Simulation and Modelling, Sharjah (United Arab Emirates), 2002–09, ISBN 90-77039-09-0

Krajzewicz D, et al (2004) Recent extensions to the open source traffic simulation SUMO. In: Proceedings of the 10th world conference on transport research (on CD), WCTR04 – 10th World Conference on Transport Research, Istanbul, Turkey

SUMO (2001–2009) SUMO web pages. http://sumo.sf.net

GNU.org. Gnu Public License. http://www.gnu.org/licenses/gpl.html. Accessed 9 Apr 2009

Blanke K, Ehling M, Schwarz N (1996) Zeit im Blickfeld: Ergebnisse einer repräsentativen Zeitbudgeterhebung. Band 121 der Reihe Schriftenreihe des Bundesministeriums für Familie, Senioren, Frauen und Jugend. Kohlhammer, Stuttgart, 1996

Hertkorn G, Kracht M (2002) Analysis of a large scale time use survey with respect to travel demand and regional aspects. International association for time use research (IATUR) – conference 2002, Lisbon

Hertkorn G, Wagner P (2004) Travel demand modelling based on time use data. 10th international conference on travel behaviour research, Luzern, 10–15 August 2004

Hertkorn G (2004) "Mikroskopische Modellierung von zeitabhängiger Verkehrsnachfrage und von Verkehrsflussmustern;" Deutsches Zentrum für Luft- und Raumfahrt; Forschungsbericht 2004–29; Köln; http://elib.dlr.de/21014/01/fb_2004-29_v2.pdf

Krajzewicz D et al (2005) Simulation of modern Traffic Lights Control Systems using the open source Traffic Simulation SUMO. In: Krüger J, Lisounkin A, Schreck, G. (eds) Proceedings of the 3rd industrial simulation conference 2005, EUROSIS-ETI, S. 299–302, 3rd Industrial Simulation Conference 2005, Berlin, Germany, ISBN 90-77381-18-X

BMWi. “Verkehrsmanagement 2010” web pages. http://vm2010.de/web/index.php. Accessed 9 Apr 2009
NAVTEQ. NAVTEQ web pages. http://www.navteq.com. Accessed 8 May 2009
PTV AG. PTV validate web pages. http://www.ptv.de/software/verkehrsplanung-verkehrstechnik/software-und-system-solutions/ptv-validate/. Accessed 8 May 2009
INVENT. INVENT project web pages. http://www.invent-online.de/. Accessed 9 Apr 2009
Ehrenpfordt I, Höpfner M, et al (2006) Systematischer Test eines innovativen Verkehrsdatenerfassungssystems. In: Deutsche Messe AG [Hrsg.]: CeBIT in Motion – Forum for Telematics and Navigation, CeBIT in Motion – Forum for Telematics and Navigation, Hannover
Höpfner M, Lemmer K, Ehrenpfordt I (2006) Validation of a GSM-based traffic monitoring system. In: ERTICO [Hrsg.]: ITS world conference 2006, 13th world congress and exhibition on intelligent transport systems and services, London, 8–12 Oct 2006
TrafficOnline. TrafficOnline project web pages. http://www.trafficonline.de/. Accessed 9 Apr 2009
Honomichl C (2008) Analyse und Simulation von Verkettbarkeitsmaßen in pseudonymer, mobiler Kommunikation, Diploma thesis. Technical University Darmstadt. 2008
Morenz T (2008) iTranSIM – Simulation-based vehicle location. Master thesis, University of Dublin
OpenStreetMap. OpenStreetMap web site. http://www.openstreetmap.org/. Accessed 14 Apr 2009
Krauß S (1998) Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics. Hauptabteilung Mobilität und Systemtechnik des DLR Köln. ISSN 1434-8454
Brockfeld E, Kühne R, Wagner P (2004) Calibration and validation of microscopic traffic flow models. In: Transportation research board (ed) TRB 2004 annual meeting, S. 62–70, TRB Annual Meeting, Washington, DC, USA
Brockfeld E, Wagner P (2004) Testing and benchmarking of microscopic traffic flow models. In: Proceedings of the 10th world conference on transport research, S. 775–776, WCTR04–10th World Conference on Transport Research, Istanbul, Turkey
Brockfeld E, Kühne R, Wagner P (2005) Calibration and validation of microscopic traffic flow models. Transportation research records, 1934, S. 179–187
Brilon W, et al (2005) Fortentwicklung und Bereitstellung eines bundeseinheitlichen Simulationsmodells für Bundesautobahnen. Forschung Straßenbau und Straßenverkehrstechnik, Heft 918. Herausgegeben vom Bundesministerium für Verkehr, Bau- und Wohnungswesen
Krajzewicz D (2009) Kombination von taktischen und strategischen Einflüssen in einer mikroskopischen Verkehrsflusssimulation, in Th. Jürgensohn, H. Kolrep (Hrsg.), Fahrermodellierung in Wissenschaft und Wirtschaft, 2. Berliner Fachtagung für Fahrermodellierung, Fortschrittsbericht des VDI in der Reihe 22 (Mensch-Maschine-Systeme), Nr.28, 104–115, VDI-Verlag, 2009, ISBN 978-3-18-302822-1
Ehmans D (2001) Simulationsmodell des menschlichen Spurwechselverhaltens; Simulation model of human lane change behaviour. In: VDI/SAE/JSAE Gemeinschaftstagung, Der Fahrer im 21. Jahrhundert, Berlin 3.-4. Mai 2001.http://www.pelops.de/publikationen
Gawron C (1998) Simulation-based traffic assignment – computing user equilibria in large street networks. PhD Dissertation, University of Köln, Germany
Behrisch M, Krajzewicz D, Wagner P, Wang Y-P (2008) Comparison of methods for increasing the performance of a DUA computation. In: Proceedings of DTA2008, DTA2008 international symposium on dynamic traffic assignment, Leuven, Belgium
Behrisch M, Krajzewicz D, Wang Y-P (2008) Comparing performance and quality of traffic assignment techniques for microscopic road traffic simulations. In: Proceedings of DTA2008, DTA2008 international symposium on dynamic traffic assignment, Leuven, Belgium
googletest. googletest web site. http://code.google.com/p/googletest/. Accessed 14 Apr 2009
TextTest. TextTest web site. http://texttest.carmen.se/. Accessed 9 Apr 2009
Wegener A, Piórkowski M, et al (2008) TraCI: an interface for coupling road traffic and network simulators. In: Proceedings of the 11th communications and networking simulation symposium (CNS′08), Ottawa, Canada

Wegener A, Hellbrück M, et al (2008) VANET simulation environment with feedback loop and its application to traffic light assistance. In: Proceedings of the 3rd IEEE workshop on automotive networking and applications, New Orleans, LA, USA

Piórkowski M, Raya M, et al (2008) TraNS: realistic joint traffic and network simulator for VANETs. SIGMOBILE Mob Comput Commun Rev 12(1):31–33. ISSN 1559-1662. doi: http://doi.acm.org/10.1145/1374512.1374522

EPFL (2008) Trans web site. http://trans.epfl.ch Accessed 05 July 2010

Schäfer R-P, Thiessenhusen K-U, Brockfeld E, Wagner P (2002) A traffic information system by means of real-time floating-car data. In: ITS world congress, 11–14 Oct 2002

Eissfeldt N (2004) Vehicle-based modelling of traffic. PhD thesis, University of Cologne, Cologne, Germany

iTETRIS. iTETRIS web site. http://www.ic-itetris.eu/. Accessed 14 Apr 2009

INFRAS. Handbook emission factors for road transport. http://www.hbefa.net/. Accessed 14 Apr 2009

Nota R, Barelds R, van Maercke D, van Leeuwen H (2005) Harmonoise WP 3 – Engineering method for road traffic and railway noise after validation and fine-tuning. Harmonoise Technical Report HAR32TR-040922-DGMR20 (Deliverable 18)

Papaleontiou L, Dikaiakos M. D. (2009) TrafficModeler: A Graphical Tool for Programming Microscopic Traffic Simulators through High-level Abstractions. IEEE 69th Vehicular Technical Conference (VTC2009-Spring), April 26–29, 2009. Barcelona, Spain

Creative Commons. Attribution-Noncommercial-Share Alike 3.0 license web site. http://creativecommons.org/licenses/by-nc-sa/3.0/. Accessed 14 Apr 2009

Chapter 8
Traffic Simulation with DRACULA

Ronghui Liu

8.1 Introduction

The DRACULA traffic microsimulation model was developed as a tool to investigate the dynamics between demand and supply interactions in road networks. The emphasis is therefore on the integrated microsimulation of individual trip makers' decisions, travel experiences, and learning. This is represented through a microscopic dynamic traffic assignment model based on the explicit modelling of individuals' day-to-day route and departure time choices and how their past experience and knowledge of the network influence their future choices. Coupled with that is a detailed within-day traffic microsimulation based on car-following and lane-changing rules.

This chapter provides an updated overview on the main functions of DRACULA model. The recent focus of our research on the issues of model calibration and validation is discussed. Some of the extended features of the software, on modelling the overtaking behaviour on two-lane rural roads and the fully integration with a microscopic model of public transport operations and demand, are represented.

8.2 Model Building Principles in DRACULA

The dynamic network microsimulation framework *DRACULA* (Dynamic Route Assignment Combining User Learning and microsimulation) has been developed at the University of Leeds since 1993. It adapted a unique approach to modelling road traffic networks whereby the emphasis is on the "microsimulation" of both individual trip makers' choices and individual vehicles' movements.

R. Liu (✉)
Institute for Transport Studies, University of Leeds, Leeds, UK
e-mail: r.liu@leeds.ac.uk

J. Barceló (ed.), *Fundamentals of Traffic Simulation*,
International Series in Operations Research & Management Science 145,
DOI 10.1007/978-1-4419-6142-6_8,

The model attempts to represent directly the behaviour of individual drivers in real time as these evolve from day to day. This is coupled with a detailed within-day traffic simulation of the continuous movements of individual vehicles according to car-following and lane-changing rules and traffic controls. In combination they model the evolution of the traffic system over a representative number of days so that both within-day and between-day variabilities are included and interaction between the demand and supply modelled.

The full DRACULA framework combines a number of sub-models. The *demand model* represents the day-to-day variability in total demand within a fixed departure time period. It simulates for each potential traveller – based on individual drivers' knowledge of the network, their past experience and perceived network condition of the day – whether to travel, if so, the route to be taken, and the preferred departure time.

This information is then passed to the *traffic simulation model* which represents the within-day variability of network conditions and simulates individual vehicle movements through the network following the routes chosen and records their travel performance.

At the end of the day (the study period), a *learning model* updates the experiences of each individual and stores the information in their travel history files which, to a greater or lesser extent, influence their next day's choices.

An overview of the DRACULA model architecture is presented in Section 8.3. Sections 8.4 and 8.5 describe the core functions in DRACULA, namely models of traffic simulation and dynamic traffic assignment. The calibration and validation of the model are presented in Section 8.6. The flexibility of the framework ensures that new modelling features can be readily incorporated into the framework, as some of the specialist features described in Section 8.6 demonstrate.

8.3 DRACULA Model Structure

The DRACULA framework integrates a number of sub-models of traffic flow and drivers' choices for a given day with a day-to-day driver learning sub-model. The overall structure of the model framework is illustrated in Fig. 8.1. Detailed discussion of the interactions of the sub-modelled can be found in Liu et al. (2006).

Briefly, the sub-models and the dynamic evolution of the demand-supply interactions they represent are as follows:

- A *population* sub-model, which synthesises the population in the study area and generates all the potential drivers (T_{IJ}^{max}) from a traditional origin–destination matrix (T_{IJ}).

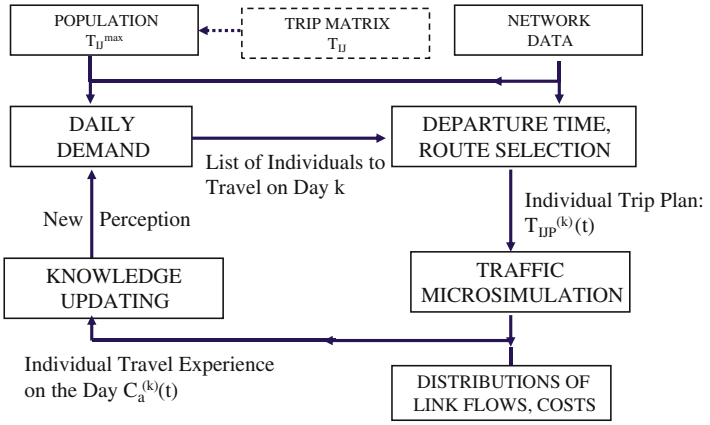
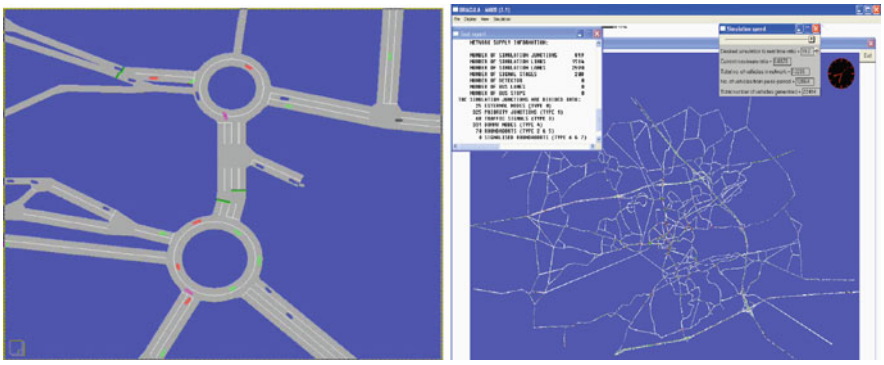


Fig. 8.1 The day-to-day evolution represented in DRACULA

- A *demand* sub-model represents the day-to-day variability in total demand. It predicts the level of individual demand for day *k* from a full population of potential drivers. At the most detailed level, this model provides the list of individual trip makers who wish to travel on the day, $T_{IJ}^{(k)}$, from their origin (*I*) to chosen destination (*J*).
- A *dynamic traffic assignment* sub-model determines the routes and departure times of the individual drivers based on their past travel experience and their perceived knowledge of the network conditions. The results are individuals' trip plans, $T_{IJP}^{(k)}(t)$, from origin *I* to destination *J*, along path *P* and depart at time *t*.
- In the *traffic microsimulation* sub-model, the individual vehicles are then moved through the network following their chosen routes according to car-following and lane-changing rules, in a traffic simulation sub-model.
- The costs experienced by drivers on the day and on each passing links, $C_a^{(k)}(t)$, are then re-entered into their individual "knowledge bases". The *learning* model would update the driver's perception of the network conditions, which in turn affects their decision for the next day *k+1*.
- A *data collection* sub-module collects measures on travel time, congestion, emission, incidents, etc.

The system evolves continuously from one day to the next until a pre-defined number of days, or a broadly balanced state between the demand and the supply is reached. Simulation results can be obtained throughout the evolution and on not just the means but also variances and probability distributions both within-day and between days.

The within-day traffic simulation is animated through a graphical user interface (GUI) which is useful both for debugging purposes and for examining the traffic impacts on network. Figure 8.2a presents a snapshot from the vehicle animation.



(a) (b)

Fig. 8.2 Illustrations of the GUI on (**a**) vehicle animation and (**b**) a large network model in DRACULA

8.3.1 Data Source and Linkage with Conventional Network Models

A microsimulation model such as DRACULA requires essentially the same basic data as traditional network models such as SATURN (van Vliet, 1982), nodes, links, number of lanes, lane markings, signal operations, and give-way rules. To this end, we have developed a direct link between DRACULA and SATURN in that the former takes the basic network and demand data from the latter.

The benefits are twofolds. On the one hand, we can take full advantage of the large data bank of the SATURN networks developed around the world, into DRACULA. Thus reducing the time it takes to code a network from scratch. Figure 8.2b shows a DRACULA model of a city in the north of England, consisting of some 200 intersections of various junction controls, 1500 links, 2600 lanes, and 142 bus routes. The model was converted *automatically* from an existing SATURN model of the network, with minimal effort.

On the other hand, users of SATURN models can run DRACULA to conduct a traffic microsimulation for detailed network design and/or short-term forecasting, with route assignment from SATURN as an exogenous input to DRACULA.

8.4 The Traffic Simulation

As with other traffic microsimulation software, the essential property of the DRACULA traffic simulation model is that the vehicles move in real-time and their space-time trajectories are determined by, e.g. car-following and lane-changing models.

However, the vehicle simulation in DRACULA interacts strongly with, and is influenced by the requirements of, its demand model. For example, the explicit

modelling of individual drivers' day-to-day learning has an impact on the length of the simulation time period (see Section 8.4.1).

The fixed within-day route choices modelled have an impact on the lane-changing model which needs to cater for the need of the drivers to follow a fixed route; therefore, choose lanes which lead to the correct turn at any particular intersection (Section 8.4.4).

8.4.1 Simulation Time Periods and Simulation Loop

The needs to associate a specific route and destination with each vehicle and to acquire the full knowledge of their complete trip from origin to destination lead to a not-specifiable simulation time period. Each simulation run has its own end time which is determined by how long it takes for all trips to complete; the latter is determined by how congested the network is.

Figure 8.3 shows the "time periods" in a typical simulation run in DRACULA. The simulation begins with a "warm-up period" prior. The main "demand period" represents a typical peak or off-peak period of a day. The individuals who wish to travel on the day as determined by the demand model would enter the network during this demand period. The simulation continues to feed vehicles into the network even when the main demand period has ended, during the "cooling-off period".

This feature allows us to generate the true cost of a trip, taking into account the full congestion impacts, as opposed to the usual engineering measure within a pre-determined time window, therefore potentially ignoring the congestion/queuing outside the time frame. A critical analysis on the differences between the cost of a trip from its origin to destination and a network performance is presented in Liu (2004) and Hill et al. (2001).

Therefore, in DRACULA, the traffic simulation ends only when all drivers from the demand period have reached their destinations. The iteration of the traffic simulation is depicted in Fig. 8.4.

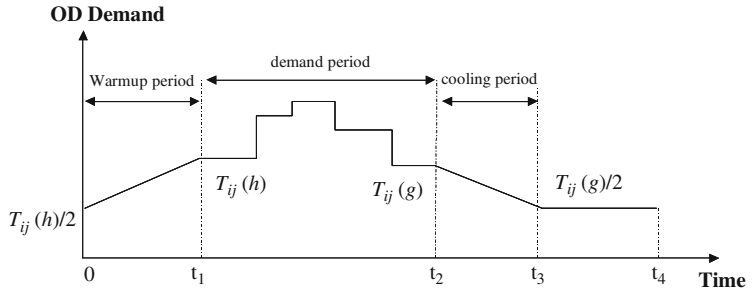


Fig. 8.3 The simulation time periods represented in DRACULA. $T_{ij}(h)$ and $T_{ij}(g)$ represents the O–D demand at the beginning and end of the main period, respectively. Time t1, t2, t3 are user-defined variables, whilst the end of simulation time, t4, is variable depending on congestion levels in networks

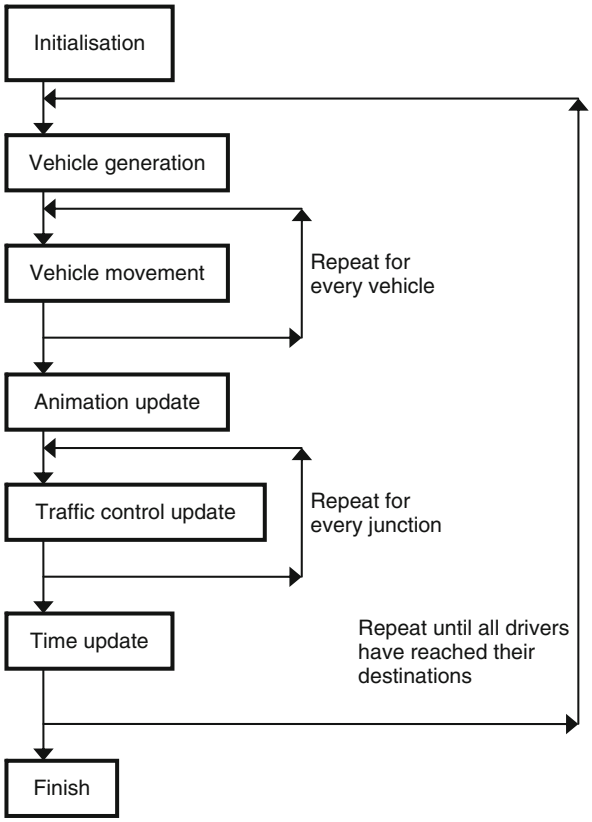


Fig. 8.4 The traffic simulation loop in DRACULA

8.4.2 Car-Following Model

The DRACULA model was originally developed for urban traffic networks, where the traffic moves in an "interrupted" state by the traffic controls at intersections. The car-following model in it was based on the car-following rules of Gipps (1981), and has been shown to be able to represent realistically the individual vehicle trajectories, and their aggregated impact on measures of saturation flow and discharge profiles, at a traffic signal controlled intersection (Liu, 2005).

The mathematical formulation of the Gipps model is reproduced in eq. (8.1), which gives the speed of each vehicle n at time $t + \tau$ in terms of its speed at the earlier time t as

$$v_n^G(t+\tau) = \min\{v_n(t) + 2.5A_n\tau[1 - v_n(t)/V_n]\sqrt{0.025 + v_n(t)/V_n},$$
$$B_n\tau + \sqrt{B_n^2\tau^2 - 2B_n[x_{n-1}(t) - x_n(t) - L_{n-1}] + B_n\tau v_n(t) + B_n v_{n-1}(t)/B'}\} \quad (8.1)$$

where $x_n(t)$ and $x_{n-1}(t)$ denote the location of following vehicle n and its preceding vehicle $n-1$ at time t, respectively, L_{n-1} the length of vehicle $n-1$, V_n the desired speed of the driver, and $A_n > 0$ and $B_n < 0$ are the acceleration and deceleration of vehicle n, respectively.

The first part of the right-hand side of eq. (8.1) represents the desire of the driver to accelerate freely to reach its desired speed V_n, whilst the second part is a constraint on the following driver to be always on the alert and to be able to bring his vehicle to a safe stop should the vehicle in front breaks to a sudden stop.

The Gipps model has the advantage that all its parameters have realistic physical meanings which make it desirable without resorting to elaborate calibration procedures. However, the speeds at which Gipps model represent are relatively low and fit mainly to the speeds of traffic usually observed on urban streets.

We present below new car-following rules developed in DRACULA to represent traffic dynamics in other situations.

8.4.3 Car-Following on Motorway Links

More recently, the traffic simulation model in DRACULA has been extended to represent the "un-interrupted" traffic flow dynamics, typically seen on high-speed, long motorway links. The new car-following model aims to capture some of the key motorway flow characteristics, namely traffic breakdown, hysteresis, shockwave propagation, and close-following behaviour. The model was fully described in Wang et al. (2005a). Here we summarise its main features.

The new car-following model was built on the concept that drivers in different traffic "conditions" behave differently. The traffic conditions considered were "traffic build-up" from free flow towards congestion, "close-following" at high speed and short headway, "traffic breakdown" as characterized by flow, and speed reductions and increasing of density, and finally "traffic recovery".

Behaviourally, the drivers are assumed to be in different alertness "states" under different traffic conditions and apply different reaction times and accelerations accordingly. The driver states modelled are "non-alert state", "alert state", and "close-following" states.

The car-following behaviour for the *non-alert* and the *alert* states is represented using the Gipps model, but with different reaction time and acceleration for the different state.

Traffic is said to be in the *close-following* state if it falls within a region of small relative speed (between ΔV_a and ΔV_b) and small relative space headway (between $d_{\min}$ and $d_{\max}$) to the vehicle in front. The model applies either a constant accelerate or a deceleration to a vehicle depending on its space headway to the vehicle in front. The speed of the following vehicle is simply updated according to Newtonian equation of motion:

$$v_n^C(t+\tau) = \begin{cases} v_n(t) + a_3\tau_3 & \text{for acceleration when } \Delta Xn \geq (\mathrm{d_{min}} + \mathrm{d_{max}})/2 \\ v_n(t) - a_3\tau_3 & \text{for deceleration when } \Delta Xn < (\mathrm{d_{min}} + \mathrm{d_{max}})/2 \end{cases} \quad (8.2)$$

where a_3 and τ_3 are the acceleration and reaction time, respectively, applied by drivers in close-following state.

The above concept of close-following was first proposed by Leutzback and Wiedemann (1986). Brackstone et al. (2002) then calibrated the model and found the boundary values that define the close-following to be

$$\Delta V_a = 2 \text{ m/s} \text{ and } \Delta V_b = -2 \text{ m/s} \tag{8.3a}$$

$$d_{\min} = L_n + C_1\sqrt{v_n(t)} \text{ and } d_{\max} = L_n + C_1\sqrt{C_2 v_n(t)} \tag{8.3b}$$

where C_1 and C_2 are constants.

At the aggregated level, the different states represented in the model and their transitions are illustrated in Fig. 8.5.

The model is shown to be able to realistically capture the key motorway traffic flow characteristics, including speed drop, and traffic hysteresis (Wang et al., 2005a). Figure 8.6 presents example results.

Figure 8.6a shows the individual vehicle trajectories simulated. Four backward shock waves can be identified with a reduction of traffic flow and velocity, i.e. where the trajectories become less condensed. The speeds of the four shock waves range between −10 and −22 m/s and are comparable to those observed on UK motorways.

One of the motivations for developing this car-following model was to model the close-following behaviour of the motorway traffic. Figure 8.6b compares the gap distribution (for gaps below 5 s) as simulated using the current model, the Gipps car-following model, and a car-following model proposed in Zhang and Kim (2001), with the observed data collected by Brackstone et al. (2002). It can be seen that the current model produced the closest match with the observation.

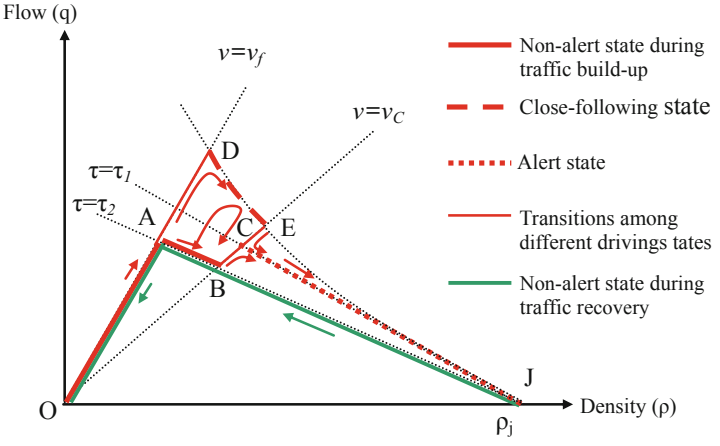
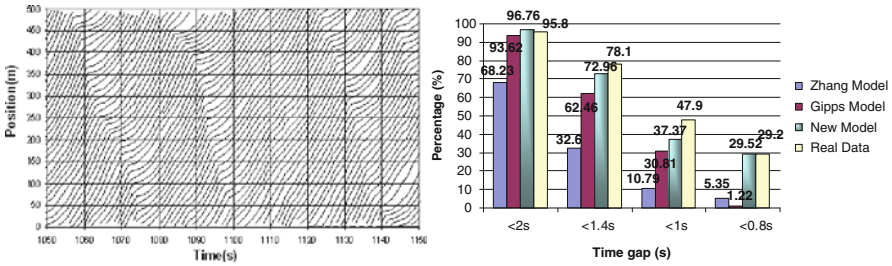


Fig. 8.5 The transitions between different driving states



(a) (b)

Fig. 8.6 Illustrations of the simulated individual vehicle trajectories (**a**) and the distributions of time gaps (**b**)

Calibration of the main model parameters was conducted by Liu and Wang (2007) and is summarised in Section 8.6.1. The stability of the proposed model was analysed in Liu and Li (2008).

8.4.4 Lane-Changing Model

The lane-changing model in DRACULA is rule based and follows a decision tree as depicted in Fig. 8.7. The whole lane-changing logics represented stems from models of two very different causes or desires for lane-changing: the vehicle is in the "wrong" lane therefore has to change lane or it wants to change lane in order to improve its "desire" or "comfort".

A lane-changing move made because the vehicle is in the wrong lane is also termed "mandatory lane-changing" where the move has to be made by a certain position on the link.

The desire to improve its speed or headway by changing to its neighbouring lane is classified as "discretionary lane-changing" which needs or needs not be carried out depending on the actual traffic conditions.

Once a lane-changing decision is made, the driver would look for the earliest opportunity, i.e. availability of gaps, to move to the target lane. A "stay-put" period is modelled after a lane-changing move when the driver is not going to attempt another lane-changing.

8.4.5 Look-Ahead Factors in Lane Changing

This is a model of drivers' looking beyond his current link to the situations or requirements of his move in the downstream link en route. If, by staying in the current lane and moving to a lane downstream lane, the vehicle would be in the "wrong lane" for that link, then a lane-changing move may be made in the current link. The factors which influence such a lane-changing decision include the following:

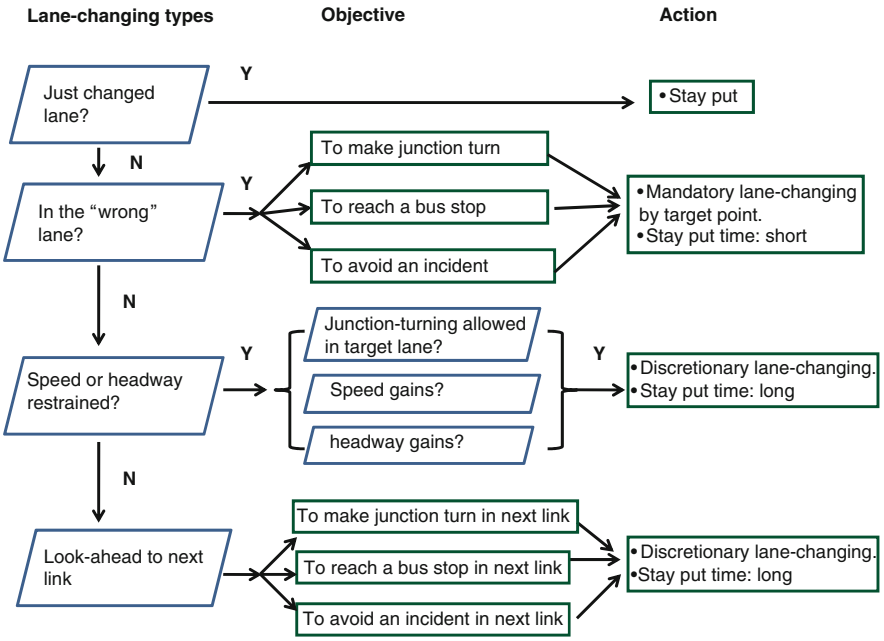


Fig. 8.7 Structure of the lane-changing decisions represented in DRACULA

- the junction-turning movement at the downstream link;
- for a bus, to reach a bus stop en route at the next link;
- if there is an incident (i.e. parked vehicle or reserved lane) at the downstream link.

As the drivers in the model follow fixed routes, the information of its next junction-turn is readily available. In the bus model in DRACULA, each bus service is associated with a pre-defined list of bus stops en route. The bus drivers are assumed to know its route well enough to know whether there is a bus stop in the next link.

8.4.6 On-Ramp Merge

Most of the lane-changing models assume that a lane-changing move is made without any direct impact on the traffic in its target lane, i.e. it would not have forced the traffic in the target lane to do anything different to what they would have done without the extra vehicle moving into their lane. There is no response or cooperation between the lane-changing vehicles and vehicles in the target lane.

In reality, there is often a kind of "cooperation" between the different streams and most notably among the two interactive traffic streams at an on-ramp motorway

merging area. There, the traffic on the mainline motorway exhibits a kind of cooperation by changing to the outer lane or by yielding, in order to create gaps for the merging traffic.

A model to represent such cooperative behaviour at motorway merge has been developed in DRACULA. The full features of the model are described in Wang et al. (2005b) and we summarise its main features here.

Figure 8.8 illustrates a typical merge area, where a merging vehicle (*C*) interacts with its putative leader (*PL*) and putative follower (*PF*) on the nearside motorway lane. The merging vehicle will examine the *original gap* between PL and PF (the first motorway gap to be faced by C when it arrives at the acceleration lane), the *previous gap* in front of PL, and the *following gap* behind PF. A PL (or PF) exists if the lead (or the lag) gap is less than 5 seconds.

The framework containing the main functions of the model is displayed in Fig. 8.8. The new merge model tries to capture the above behaviour through a number of sub-models described as follows:

(a) Cooperation model. Cooperative yielding behaviour and cooperative lane-changing are both modelled as a random decision made by the PF as to whether or not to reduce its speed to create gaps for C or to move to the offside lane(s).
(b) An acceleration model. This models the acceleration or deceleration of C towards its target gap whilst maintaining a safe distance away from the vehicle in front in the acceleration lane.
(c) A gap selection model. Based on its speed and location relative to its PF and PL, the driver of vehicle C will select a target gap to merge.
(d) A gap-acceptance model. Here the acceptable lead and lag gaps are calculated as a function of the speed; merging driver's reaction time; and maximum deceleration of vehicle C, PF, and PL by considering the forecast of their actions in the merging process.
(e) A merge model. When an acceptable gap is found, vehicle C merges into the motorway traffic. However, if the vehicle has not found an acceptable gap before reaching the end of the acceleration lane, a merge failure will be registered.

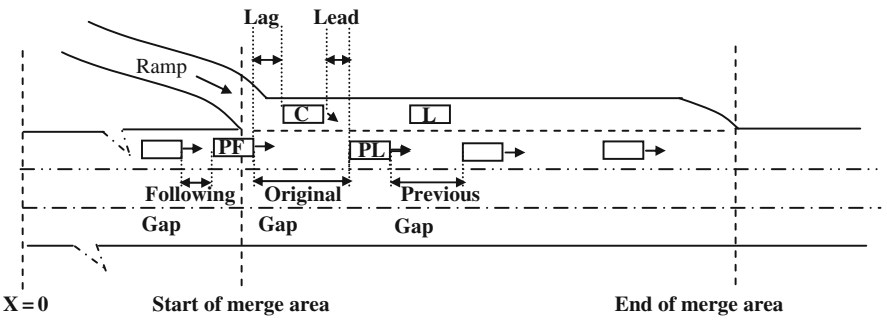


Fig. 8.8 A schematic diagram of a merge area

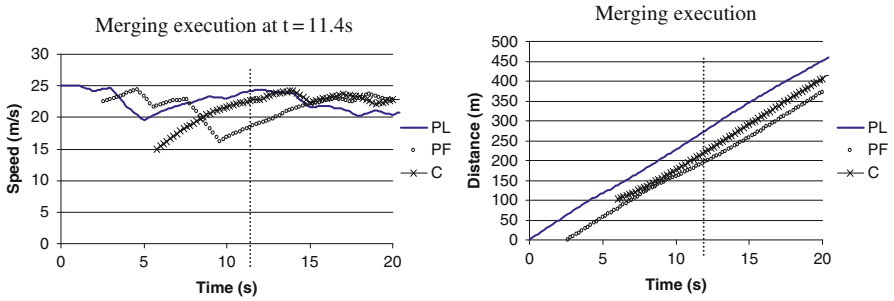


Fig. 8.9 Speed–time and distance–time profiles of vehicles involved in a merge under courtesy yielding

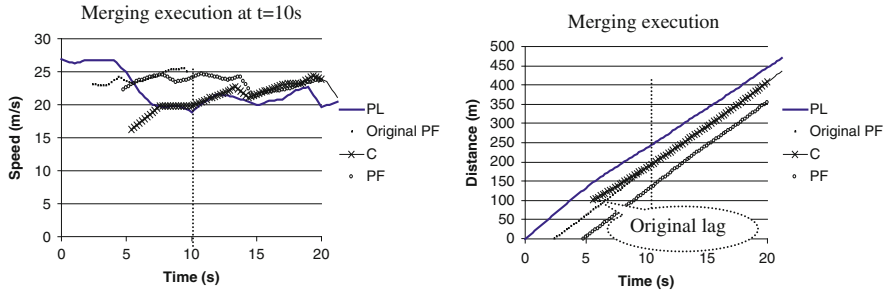


Fig. 8.10 Profiles of the vehicles at merge with a cooperative lane changing

Figs. 8.9 and 8.10 show example simulation results which illustrate the ability of the model to replicate a merging under courtesy yielding and cooperative lane-changing, respectively.

In Fig. 8.9, one can see that just before the merging took place at $t = 11.4$ s, the PF vehicle reduced its speed (Fig. 8.9a) which led to larger and acceptable lag gap (Fig. 8.9b) for vehicle C to merge.

Figure 8.10 shows traces of an original PF before the merging took place at time $t = 10$ s and the lag to the original PF was small (Fig. 8.10b). When the original PF made a cooperative lane-changing move (whose traces are therefore disappeared from the diagrams), the lag gap to the new PF became acceptable to vehicle C to merge.

8.5 Dynamic Traffic Assignment

DRACULA is developed as a tool to test fundamental properties of dynamic traffic assignments (DTA). To this end, various DTA sub-models have been implemented and studied in DRACULA. These models vary by details and behavioural assumptions.

At its most detailed level is the day-to-day, microscopic model of individuals' route and departure time choices, and an individual-based learning model. Section 8.5.1 summarises its main functions, whilst more details on this model can be found in Liu et al. (2006).

At a more aggregated level, a simple dynamic route choice model based on the aggregated response to overall system performances was also implemented in DRACULA. Section 8.5.2 discusses some of the issues related to this type of models.

In practice, DRACULA can also be linked to an exogenous traffic assignment models and be used as a pure traffic microsimulation, an example of which is illustrated in Section 8.3.1.

All the route choice models represented in DRACULA are pre-trip DTA; there is no en route route choice currently represented in DRACULA.

8.5.1 The Full, Day-To-Day DTA Model in DRACULA

In this version of the DTA model, each individual is represented. Their daily route choices are explicitly modelled and are based on each individual's past experience and perception of the network.

Two possible route choice models are implemented in DRACULA: (a) "bounded rational": stay on habit route unless the alternative is well better and (b) myopic switch: always take the least cost route.

The bounded rational model, based on the work of Mahmassani and Jayakrishnan (1991), assumes that drivers will use the same (habit) route as on the day they last travelled, unless the cost of travel on the minimum cost route is significantly better. The threshold is that a driver will use his habit route unless

$$C_{p1} - C_{p2} > \max(\eta \times C_{p1}, \varphi) \tag{8.4}$$

where C_{p1} and C_{p2} are costs along the habit and the minimum cost routes, respectively, and η and φ are global parameters representing the relative and absolute cost improvement required for a route switch.

The myopic route choice is a special version of the above in which the threshold is zero, i.e.

$$C_{p1} - C_{p2} > 0 \tag{8.5}$$

Here, a driver would always take the better alternative route.

The above two route choice models were implemented at the individual level, i.e. C_{p1} and C_{p2} are the habitual and minimum route costs for an individual. These individual costs were updated from their own past travel experience, through the day-to-day individual learning model.

Following the completion of trip (k−1), the perceived cost of the driver on link a would be a weighted average of costs incurred in the previous k ($<N$) trips, as

$$C_a^{(k)} = f(\lambda)\{C_a^{(k-1)} + \lambda C_a^{(k-1)} + ... + \lambda^{N-1} C_a^{(k-N)}\} \tag{8.6}$$

where $f(\lambda)$ is a scaling factor to make the weights sum to unity, and N is a modelled parameter, representing the maximum number of remembered experience.

Another parameter, M, is introduced to the model to represent the memory length where experiences more than M days old are forgotten. Thus, it is expected that $N < M$.

Liu et al. (2006) represented some of the general properties of the above day-to-day DTA model.

8.5.2 A Simple DTA Model

The simple model is based on the calculation of the route choice probabilities from given costs on alternative routes. This model assumes that the route choice proportions $\boldsymbol{P}$ at given route costs $\boldsymbol{t}$ is made according to a choice model with a dispersion parameter α>0:

$$P_{ijk}^{n+1} = \frac{(C_{ijk}^n)^{-\alpha}}{\sum_{l=1}^{m} (C_{ijl}^n)^{-\alpha}} \tag{8.7}$$

where C_{ijk}^n is the cost along route k from origin i to destination j in day n, m the number of routes used in day n for OD pair ij, and P_{ijk}^{n+1} the proportion of trips from the OD pair (ij) along route j on the following day (n+1).

The parameter α represents the degree of heterogeneity in drivers' route selection and is used to "disperse" drivers among alternative routes for a given OD pair. The higher the value of α, the more homogeneous the drivers are in their route selection behaviour.

This type of models can be found implemented in other traffic microsimulation software packages (e.g. VISSIM). The main attraction of it is its simplicity. However, there are a number of intrinsic problems with such type of "dynamic" route choice models.

First, the route choice depends on the last day's costs only. Though it is possible for such a model to converge to a deterministic user equilibrium solution where all drivers are on minimal and equal cost routes, this is not achieved naturally if they are all aiming to minimise the same travel cost by the same adjustment process.

Hazelton and Watling (2004) have shown that it is only possible to achieve equilibrium at low demands and high degree of heterogeneity in drivers' route choice. At higher value of α, which represents greater homogeneity in drivers' route selection behaviour, the system can be attracted towards a "flip-flopping" flow behaviour, whereby the route flow split alternate between two states from one day to another. This behaviour occurs mostly at high demand, e.g. at the steeper parts of the supply curves.

In reality, drivers don't simply rely on their last day's experience and forget all previous experience. A more appropriate representation would be to take into account the previous N days' experience and to use some weighted factors to represent the relative significance of the N days' experience. This may help damp the flip-flopping between alternative routes (Horowitz, 1984).

Second, when α values are smaller, drivers become less dependent on the actual route costs and their perceived costs become more random until a point where the system converges to a solution whereby trips become more equally distributed among alternative routes. This will be a stable solution, but not necessarily the optimal one.

8.6 Model Calibration and Validation

Microsimulation models employ a large number of parameters to represent the complex in driver behaviour and system controls. Bonsall et al. (2005) reviewed the behavioural assumptions made, the sources and the values used for key parameters in traffic microsimulation models, and conducted sensitivity analysis on some of the model parameters using DRACULA. They demonstrated that the model predictions – and perhaps policy decisions – are sensitive to the value of some of the key parameters and that central to the success of any application of microsimulation models is a credible model suitably calibrated and validated for the application purpose and the available data source.

Model calibration is a process whereby the values of model parameters are adjusted so as the modelled output matches, or are comparable with, real-world observation. Data from a different time period or from a different site can be used in the validation process, using the calibrated parameter values, in which measures of goodness-of-fit are used to quantify the similarity between the observed and simulated data.

Methodologies for calibrating traffic microsimulation models have been proposed in many recent publications (e.g. Dowling et al., 2004; Hourdakis et al., 2003; Toledo et al., 2003). Hollander and Liu (2008a) conducted an extensive review on the literature on the methodologies for calibrating traffic microsimulation models.

This section presents example calibration and validation exercises made on the DRACULA model, using various different calibration data sources and under different application scenarios.

8.6.1 Calibrating Car-Following Models on Open Highway

Liu and Wang (2007) proposed a generic methodology to calibrate car-following models on an open highway. An open highway network is characterised by traffic coming in from one end and leaving from the other. This is opposed to traffic

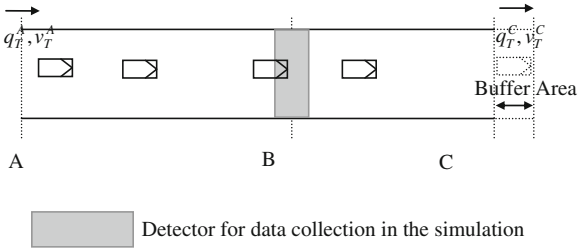


Fig. 8.11 Simulation configurations on an open stretch of highway

traversing in a loop, often used in testing car-following models (e.g. Zhang and Kim, 2001).

In calibrating a car-following model on an open highway, Liu and Wang suggested to use at least three detectors along the study section. A schematic drawing of the relative locations of the three detectors is shown as detectors A, B, and C in Fig. 8.11. The data from the upstream detector (A) is used to generate the input traffic; data from the detector located at the end of the section (C) will be used to constrain the outflow traffic, whilst data collected from the detector in the middle (B) will be used for model calibration and validation.

The method employs the most readily available traffic surveillance data: the loop detector data on average traffic speed and flow. The calibration is formulated as an optimization problem which seeks to minimize the discrepancy between the observed and the modelled traffic flow and speed. The objective function for the optimisation is formulated as

$$\min F = \sum_{\{\beta\}} \sum_{t} \left[\left(\frac{v_t^{\text{sim}} - v_t^{\text{obs}}}{v_t^{\text{obs}}} \right)^2 + \left(\frac{q_t^{\text{sim}} - q_t^{\text{obs}}}{q_t^{\text{obs}}} \right)^2 \right] \tag{8.8}$$

where $\{\beta\}$ is the set of parameters to be calibrated, t the aggregating time interval, v_t^{sim} and v_t^{obs} the simulated and observed speeds, respectively, in time period t, and q_t^{sim} and q_t^{obs} the simulated and observed flows, respectively, in t.

An automatic, iterative procedure is then carried out to find the best set of parameter values which minimise F. Figure 8.12 illustrates the solution algorithm for the calibration process.

The method was applied in the calibration of the car-following model of motorway traffic described in Section 8.4.3. The 1 min average speed and flow data from detectors on the M25 motorway in the UK, collected from the MIDAS system, was used to demonstrate the methodology. Two sets of model parameters, $\{\beta_1\} = \{a_1, \tau_1\}$ and $\{\beta_2\} = \{a_2, \tau_2\}$, representing the acceleration (a) and reaction time (τ) of the drivers during the alert and non-alert state, respectively, were calibrated.

Figure 8.13 shows an example output from the calibration process. An optimal set of the calibration parameter values at $\{\beta_2\} = \{a_2, \tau_2\} = (1.6\ \text{m/s}^2, 1.4\ \text{s})$ have been found.

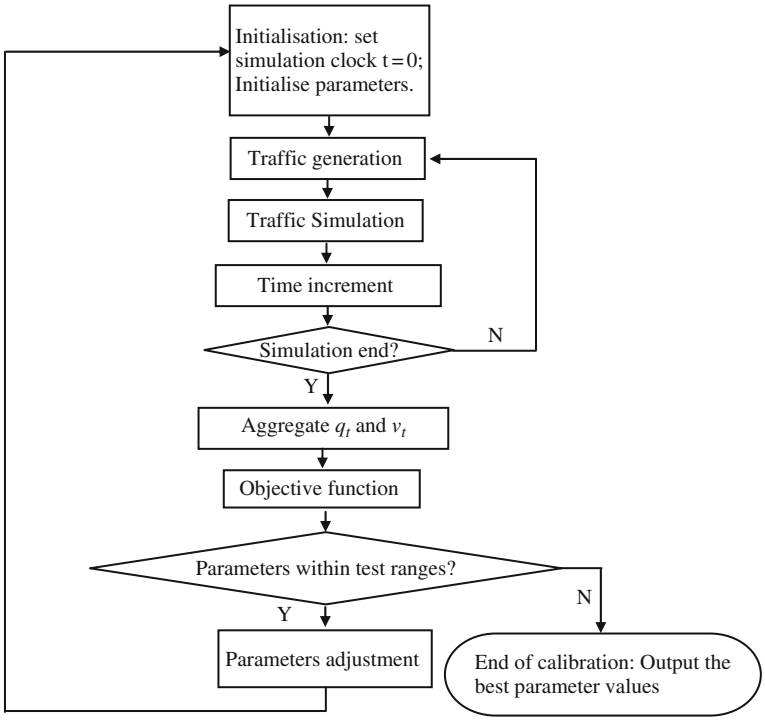


Fig. 8.12 An automatic solution algorithm

8.6.2 Calibration and Validation of the Motorway Merge Model

The merge model described in Section 8.4.6 was calibrated against observed data from video recordings of 2 h merging traffic at a UK motorway. The lead and lag gaps of each individual merges, and the number of merges took place were recorded and extracted (Zheng, 2002).

The calibration was formulated as an optimization problem which seeks to minimize the difference between the observed percentages of successful merges using the original gap and that simulated as

$$\min P = \sum_{\{\beta\}} (P^{\text{sim}} - P^{\text{obs}}) \tag{8.9}$$

where $\{\beta\}$ is the set of parameters to be calibrated, P^{sim} and P^{obs} the simulated and observed percentage of successful merges.

The detailed calibration procedure and the results of the calibration can be found in Wang et al. (2005b).

To validate the model, the calibrated parameter values were applied in the model. The modelled accepted lead and lag gaps were compared with those observed.

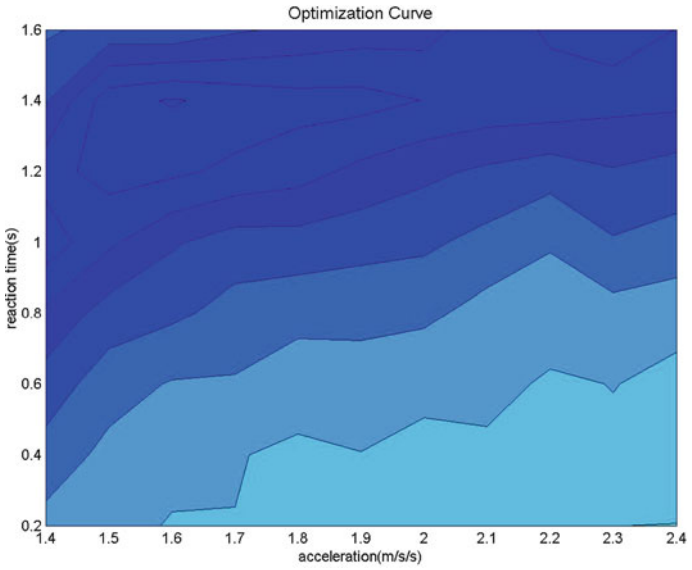


Fig. 8.13 The contour plot of the optimisation process with respect to calibrating drivers' acceleration and reaction time during non-alert state

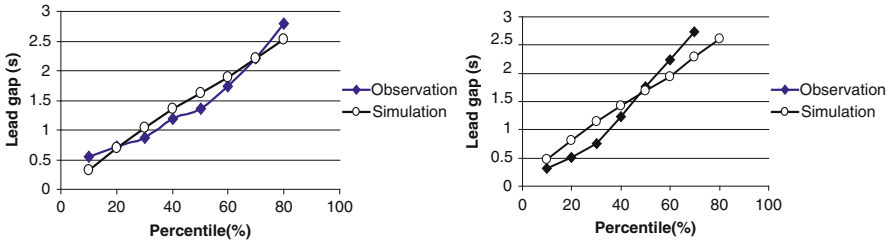


Fig. 8.14 Cumulative distributions of the lead and lag gaps

Figure 8.14 compares the cumulative distributions of the simulated and the observed lead and lag gaps. A regression of the individual percentiles gives R^2 values at 0.936 and 0.965 for the lead and lag gaps, respectively, which indicates that the distributions of the simulated lead and lag gaps compare well with those observed.

Figure 8.15 displays the simulated individual accepted lead and lag gaps as a function of the relative speeds between the merging vehicle and its PF and PL on the motorway compared to a minimum gap thresholds derived from the observed data (Zheng, 2002). One can see that most of the accepted gaps from the proposed merge model are above the thresholds.

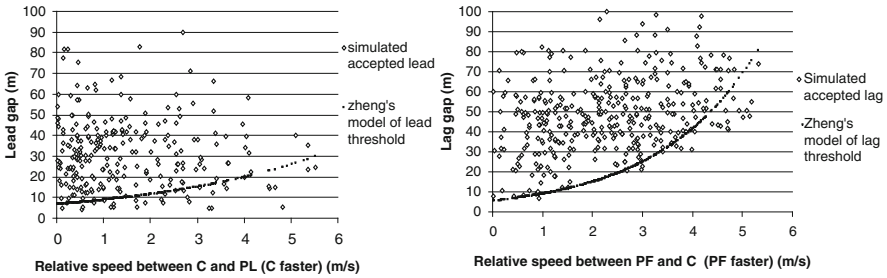


Fig. 8.15 The simulated lead and lag gaps and the observed thresholds

8.6.3 Calibration for the Distribution of Travel Time

Most of the discussions on the calibration of traffic microsimulation models in the literature have so far been concerned with calibrating the *mean* values of model outputs (e.g. mean travel times). Hollander and Liu (2008b) proposed a methodology to calibrate traffic microsimulation models to not just the mean but also the variance of travel times, i.e. the *distribution* of travel times.

They formulated the calibration process as an optimization process, which minimises the difference between the observed and the simulated cumulative probability density curves of the travel times based on the Kolmogorov–Smirnov (K–S) test. The objective function was formulated as the average of K-S values over all measurements:

$$\min Z = \frac{1}{N_{\mathrm{LP}}} \sum_{l} \sum_{p} \left\{ \max \left| F^{\mathrm{obs}}_{l,p,t} - F^{\mathrm{sim}}_{l,p,t} \right|_t \right. \tag{8.10}$$

where Z is the value of the objective function, N_{LP} the number of combined location and time-period measurements, $F_{l,p,t}$ the cumulative probability density of travel time at time t, location l, and period p. Function $\{\max \mid x \mid_t\}$ represents the maximum value of x over t.

The proposed calibration methodology was applied to calibrate 21 model parameters in DRACULA. A modified Downhill Simplex method was used to search optimal solutions for such a multi-dimensional minimisation problem.

The experiment was conducted on a test network representing a section in the City of York. Figure 8.16 shows the test network where the path along points 1–4 is a main bus route through the network with bus stops are located at these points. The travel times between bus stops, on route segments 1–2, 2–3, and 3–4, were used in the calibration experiment.

Three scenarios, with the same mean travel times but different travel time variances, were tested. The model parameter tests, their initial values, and the calibrated values for each of the scenarios are summarised in Table 8.1.

We can see that there is a reasonable level of consistency between the different test scenarios, which indicates that the calibration procedure was successful

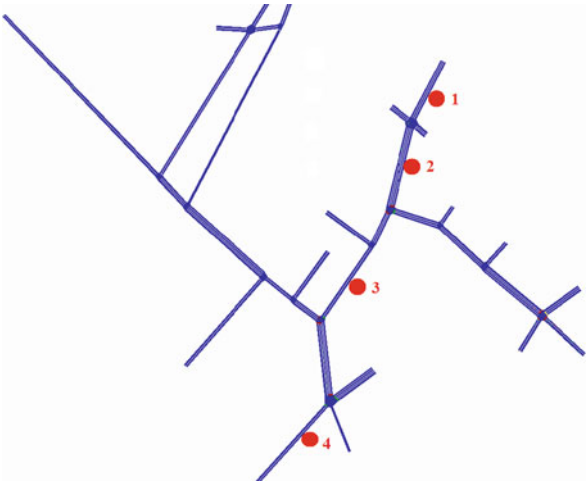


Fig. 8.16 The test network. The *numbers* and *solid circles* indicate the bus stops

Table 8.1 Summary of the calibration experiment

		Calibrated value for different coefficient of variation (cov) in travel time		
Model parameter	Initial value	0.1	0.17	0.25
Normal acceptable gap (s)	3.0	3.15	2.39	2.99
Minimum acceptable gap (s)	0.5	0.84	0.61	1.75
Waiting time 1 in gap function (s)	30.0	30.43	31.96	37.52
Waiting time 2 in gap function (s)	60.0	47.56	68.52	50.92
Demand fluctuation (cov of overall demand)	0.0	0.04	0.06	0.08
Car normal acceleration (mean, cov)	(1.5, 0.1)	(4.41, 0.22)	(2.48, 0.20)	(4.28, 0.07)
Car max. acceleration (mean (m/s^2), cov)	(2.5, 0.1)	(3.43, 0.11)	(3.99, 0.20)	(3.88, 0.17)
Car normal deceleration (mean (m/s^2), cov)	(2.0, 0.1)	(2.12, 0.03)	(3.63, 0.18)	(3.77, 0.15)
Car max. deceleration (mean (m/s^2), cov)	(5.0, 0.1)	(4.75, 0.22)	(4.53, 0.18)	(4.51, 0.17)
Bus normal acceleration (mean (m/s^2), cov)	(1.5, 0.1)	(1.86, 0.08)	(1.57, 0.18)	(1.64, 0.24)
Bus max. acceleration (mean (m/s^2), cov)	(1.6, 0.1)	(0.9, 0.24)	(1.14, 0.16)	(1.82, 0.11)
Bus normal deceleration (mean (m/s^2), cov)	(1.5, 0.1)	(1.61, 0.06)	(2.63, 0.07)	(1.35, 0.21)
Bus max. deceleration (mean (m/s^2), cov)	(2.5, 0.1)	(2.25, 0.27)	(2.87, 0.12)	(0.48, 0.20)

in preventing the model from rendering very biased estimates (Hollander and Liu, 2008b).

It should be noted that the experiment summarised above was conducted as a proof-of-concept of the calibration methodology proposed. The results of the calibration, although reasonable, were not validated with empirical data.

8.7 Extended Modelling Capabilities and Advanced Applications

8.7.1 Overtaking on Two-Lane Rural Roads

A specialist feature in DRACULA is the model of rural roads with a single lane on each direction and the overtaking behaviour on such roads which takes place using the highway in the opposite direction.

Each link in a rural road network is specified as either having "double white lines" or not having them to represent whether overtaking movements using opposite road space are generally possible or not.

A vehicle would start an overtaking move using the road space in the opposite direction if, and only if, all of the following conditions exist:

(1) The vehicle's speed is constrained by a slower moving vehicle in front;
(2) The vehicle is approaching a slow-moving vehicle in front;
(3) There is enough gap in front of the preceding vehicle for it to merge back. An assumption is made here to allow overtaking of one vehicle at a time, not a platoon of vehicles. The assumption is valid for the A614 rural network studied since traffic flow is generally low.
(4) Gap in the opposing traffic is large enough for a safe overtaking;
(5) There is clear sight distance in front.

The model structure is in general agreement with the Australian Road Laboratory's rural highway model TRARRS (McLean, 1989). The model parameters are calibrated against field observations as described below.

A rural network covering a 5.9 km section of the rural road A614 between Howden and Holme-Upon-Spalding Moor in North Yorkshire is modelled (Fig. 8.17). In the model, there is just one route with origins and destinations at both ends. The side roads, which have very small traffic in/out of them, are not represented.

Road sections where overtaking was possible was identified and was specified in the model. Similarly, the observed traffic counts at the two ends of the network, the percentage of HGVs, and vehicles' free-flow speeds were also inputs to the model.

To calibrate the model two half-hour registration plate surveys were carried out on vehicles entering and leaving the network. This data was used to calculate the average and distribution of journey times of vehicles, and the number of overtaking

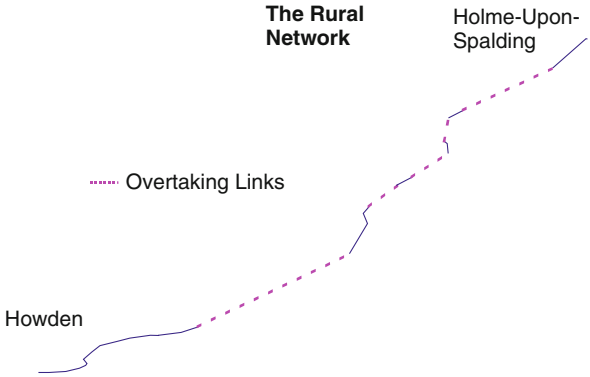


Fig. 8.17 Link properties of the modelled network. The *dashed* sections represent links where overtaking is permitted, whilst in *solid* sections they are not

Table 8.2 Journey times and number of overtaking in the network

	Observed		Modelled	
	Journey time (s) (mean, st. dev.)	Av. no. of over-takings	Journey time (s) (mean, st. dev.)	No. of overtaking mean (min, max)
Whole network	(278, 35)	50	(282, 40)	51 (32, 75)
Overtaking link	(61, 9)	–	(60, 10)	–

manoeuvres that occurred in the period. The model was run 50 times and the results were shown to be comparable with those observed (Table 8.2).

It was expected that the observed journey times would be slightly lower than the actual journey times due to drivers reacting to the presence of the surveyors on street.

During the survey 50 overtaking manoeuvres were counted. As the flows on the road are quite low, the number of overtaking manoeuvres is likely to vary from hour to hour as the composition of the vehicle fleet varies. This is supported by the results of the model, which produce varying numbers of overtaking manoeuvres for different runs, from 32 to 75 with the average being 51, the standard deviation 11, and the 95% error ± 3.

8.7.2 Integrated Highway and Public Transport Network Model

DRACULA is a microscopic traffic simulation model. It has recently been further developed to enable real-world bus operations to be represented, showing how congestion affects operations and providing an assessment tool for looking at changes to the network, bus priority measures, etc. In addition, the integrated model has the capability to represent passengers' route choice and their impacts on bus capacity and dwell time.

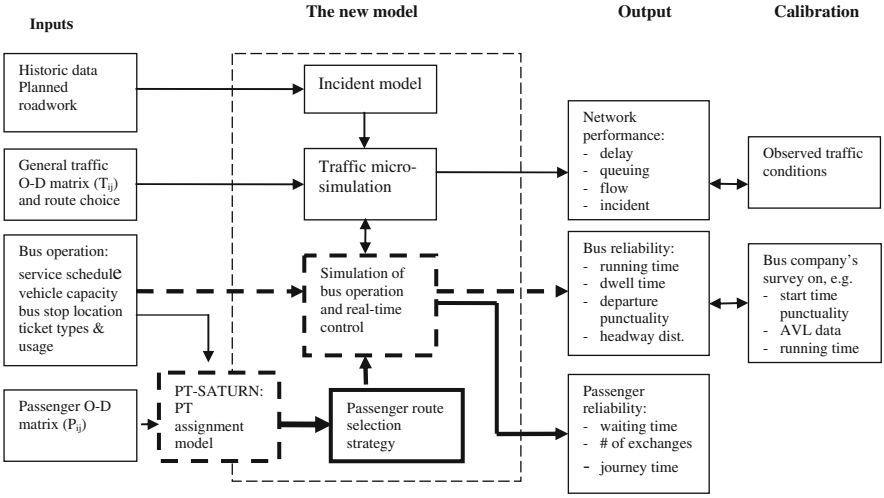


Fig. 8.18 The structure of the integrated network model

Under a UK Department for Transport and UK engineering research council-funded project, the DRACULA model was extended to represent realistically public transport operations, passenger demand and route choice, and microsimulation of the movements of individual vehicles (cars and buses) and passengers in a road network. The extended models were fully integrated with the traffic microsimulation in DRACULA. Figure 8.18 shows the integrated model framework.

This extended, integrated highway and public transport network microsimulation model has the procedures capable of

(i) realistic representations of bus operations: bus capacity, timetable, real-time controls, ticketing systems, and dwell time;
(ii) passenger demand and passenger flow through a network with explicit models of passengers route choice (including interchanges), walking (to/from bus stops), transfers (between bus stops), boarding, and alighting;
(iii) measures of bus reliability as seen from the standard point of both the service operators and the passengers, including outputs on: travel time reliability, headway reliability, punctuality (as deviation to scheduled departure time, arrival time, and timetable), and passenger excess wait time.

The model offers a tool to evaluate proposed public transport priority measures, management and control strategies and infrastructure changes and to assess their effect and the effect of congestion on service performance such as reliability. The model would be suitable for the analysis of issues such as the following:

- The effects of day-to-day and within-day variability in traffic congestion on service reliability;

- The effects of passenger demand and routing on service operation;
- Real-time service planning and operation in response to network congestion and passenger demand;
- The impact of service reliability and coverage on passenger delay and accessibility.

The integrated model was applied in a case study of bus reliability in the City of York, England. Empirical and DRACULA microsimulation study of a section of the Route 4 bus service in York (boxed section in Fig. 8.19a was carried out to quantify the causes of unreliability and to suggest measures to improve reliability.

The DRACULA model of the study area has been calibrated against observed average journey times between bus stops (Fig. 8.20) and the day-to-day variability in these journey times (Fig. 8.21).

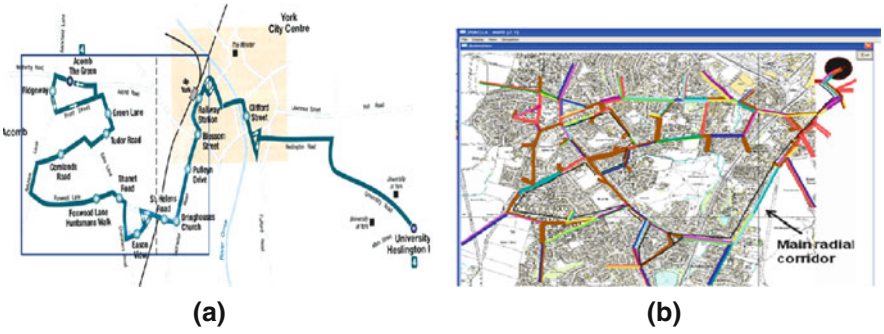


Fig. 8.19 Bus Route 4 in York (**a**) and the modelled network (**b**)

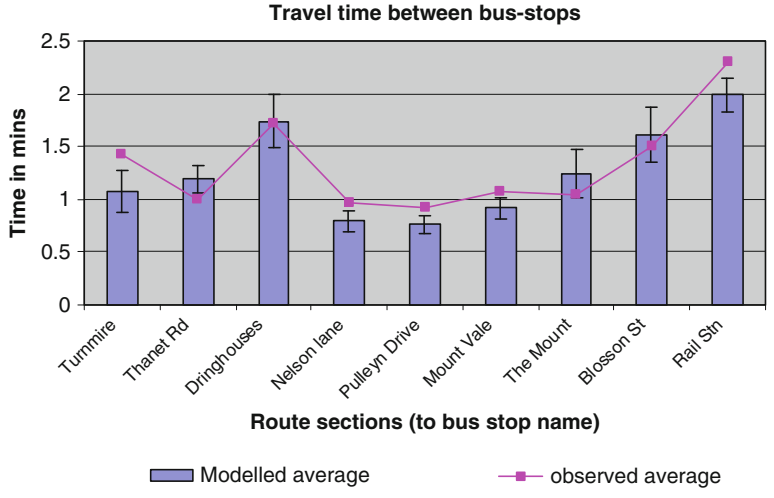


Fig. 8.20 Observed and modelled average journey times between bus stops

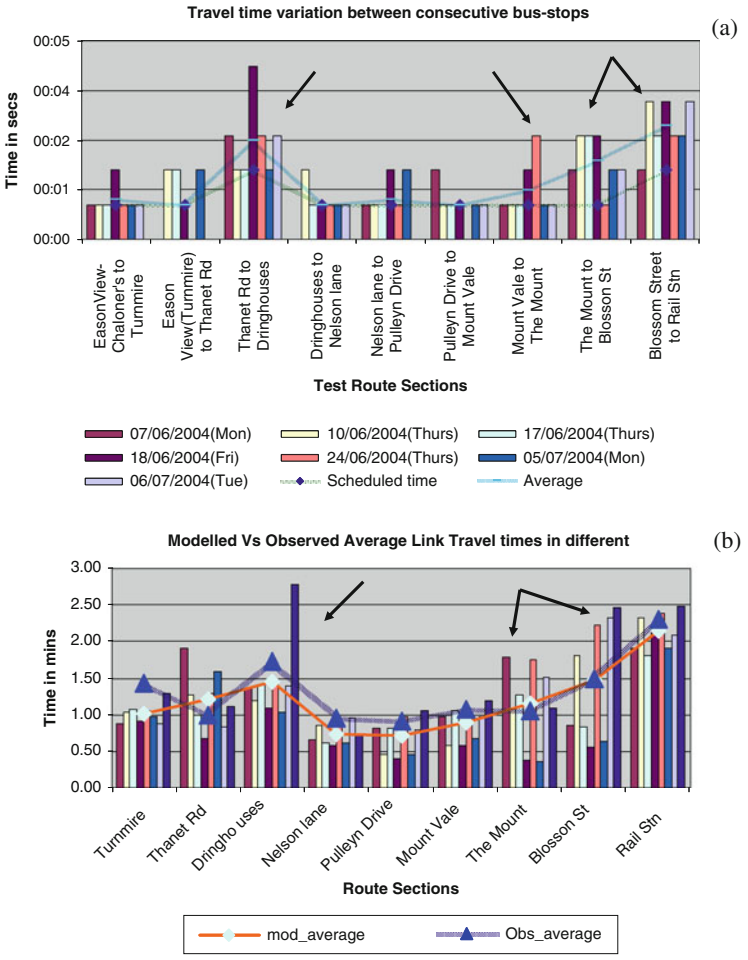


Fig. 8.21 Variability in journey times from the observation (**a**) and the simulation (**b**). The *arrows* point to the sections where large variability was observed and reproduced realistically by the simulation model

Figure 8.20 shows the modelled average journey times between bus stops; the results compare very well with those observed. Figure 8.21a shows the between-stop journey time over seven observed days. It can be seen that there is large day-to-day variability in journey times, especially along the sections of the route marked by the arrows. The model was able to reproduce the level of day-to-day variability and at the correct locations (Fig. 8.21b).

Three reliability indicators were selected in the study to represent travel time reliability, headway reliability and passenger wait time reliability. Their definitions are given in Table 8.3.

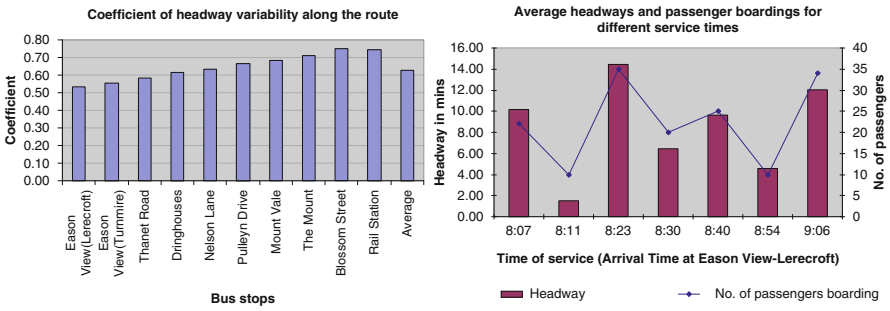
Table 8.3 Definitions of bus reliability measures

Reliability indicator	Definition	Description
Travel time reliability	$RT_i = \delta_{ti}/\mu_{ti}$ for $t_i \in \{t_{inm}\}$	Normalised dispersion of route travel time
Headway reliability	$RH_i = \delta_{hi}/\mu_{hi}$ for $h_i \in \{h_{inms}\}$	Normalised dispersion of headway en route
Passenger wait-time reliability	$RW_{is} = AWT_{is} - SWT_{is}$	Difference between actual wait time (AWT) and scheduled wait time (SWT)

t for journal time, h headway, δ standard deviation, μ mean for route i, over n bus trips, s stops, m simulation runs.

The simulation scenarios and study results were reported in Liu and Sinha (2007) and are summarised below:

- Unreliability increases with increasing congestion and passenger demand;
- The impact of rises in passenger demand was more serious on bus headway variability and passenger excess wait time, than on average bus travel time;
- Reducing boarding time per passenger brings significant improvement in reliability;
- Extension of bus lane without priorities at junctions does not yield the desired reliability benefits;
- Day-to-day variability in bus journey times was high (Fig. 8.21);
- Bus headway variability increased with route length (Fig. 8.22a);
- Headway variation and passenger boarding interrelated (Fig. 8.22b);
- Excess passenger wait time was highly dependant on variations in headways.



(a) (b)

Fig. 8.22 Headway reliability en route (**a**) and the correlation between headway variation and the number of passenger boarding (**b**)

8.7.3 Summary

This section presented just two of the extended capabilities of the DRACULA model suite and their applications. As a research tool, new capabilities and applications are continuously being implemented in and test with DRACULA. The modular structure of the framework offers the flexibility that new sub-models, or different model behaviour, can be incorporated or modified readily within the overall framework.

It is also worth noting that the fully integrated urban public transport and highway traffic network model, as presented in Section 8.7.2, offers many opportunities for application. It enables real-world bus operations to be represented, showing how congestion affects operations and providing an assessment tool for looking at changes to the network, bus priority measures, etc. As well as modelling bus operations, it has the potential to represent passengers as well as buses, the additional information adding sophistication to the analysis that can be undertake

Acknowledgements The author is grateful to Jiao Wang, Yaron Hollander, and Shalini Sinha for their contributions made whilst carrying out their studies at the University of Leeds.

References

Bonsall P, Liu R, Young W (2005) Modelling safety-related driving behaviour – impact of parameter values. Trans Res 39A:425–444

Brackstone M, Sultan B, McDonald M (2002) Motorway driver behaviour: studies on car following. Trans Res 5F:31–46

Dowling R, Skabardonis A, Halkias J, McHales G, Zammit G (2004) Guidelines for calibration of microsimulation models: framework and applications. Trans Res Record 1876:1–9

Gipps PG (1981) A behavioural car-following model for computer simulation. Trans Res 15B: 105–111

Hazelton M, Watling DP (2004) Computation of equilibrium distributions of Markov Traffic-Assignment Models. Trans Sci 38(3):31–342

Hill P, Liu R, May AD, Schmocker J-D, Shepherd S (2001) Characterising congestion on urban road networks. Park C-H (ed) Selected proceedings of 9th world Conference on transport research, vol 2. Elsevier, Seoul, 2001

Hollander Y, Liu R (2008a) The principles of calibrating traffic microsimulation models. Transportation 35(3):347–362

Hollander Y, Liu R (2008b) Estimation of the distribution of travel times by repeated simulation. Trans Res 16C:212–231

Horowitz JL (1984) The stability of stochastic equilibrium in a two link transportation network. Trans Res 8B:13–28

Hourdakis J, Michalopoulos PG, Kottommannil J (2003) A practical procedure for calibrating microscopic traffic simulation models. J Trans Res Board No. 1852:130–139

Leutzbach W, Wiedemann R (1986) Development and applications of traffic simulation models at the Karlsruhe Institut Fur Verkehrwese. Traffic Eng Control 27, 270–278

Liu R (2004) Analysis of congested networks. Proceedings of 5th Triennial Symposium on Transportation Analysis, Guadeloupe, June, 2004

Liu R (2005) The DRACULA microscopic traffic simulation model. Kitamura R, Kuwahara M (eds) Transport simulation, Springer, New York

Liu R, Hyman G (2008) Towards a generic guidance for modelling motorway merge. Paper presented at ETC Conference, Leiden, Oct 2008

Liu R, Li X (2008) Stability of a car-following model: an engineering analysis. Paper presented at UTSG annual conference, Southampton, January 2008

Liu R, Sinha S (2007) Modelling urban bus service and passenger reliability. Paper presented at the international symposium on transportation network reliability, The Hague, July 2007

Liu R, van Vliet D, Watling D (2006) Microsimulation models incorporating both demand and supply dynamics. Trans Res 40A:125–150

Liu R, Wang J (2007) A general framework for the calibration and validation of car-following models along an uninterrupted open highway. In: Heydecker B (ed) Mathematics in transport, pp 111–124

Mahmassani HS, Jayakrishnan R (1991) System performance and user response under real-time information in a congested traffic corridor. Trans Res 25A:293–308

McLean JR (1989) Two-lane highway traffic operations: theory and practice. Gordon and Breach Publishers, New York

Toledo T, Koutsopoulos HN, Davol, A Ben-Akiva ME (2003) Calibration and validation of microscopic traffic simulation tools: Stockholm case study. J Trans Res Record 1831:65–75

Van Vliet D (1982) SATURN – a modern assignment model. Traffic Eng Control 23(12):578–581

Wang J, Liu R, Montgomery FO (2005a) A car following model for motorway traffic. J Trans Res Record 1934:3–42

Wang J, Liu R, Montgomery FO (2005b) A simulation model of motorway merging behaviour. In: Mahmassani H (ed) Transportation and traffic theory: flow, dynamics and human interaction, pp 281–302, Elsevier, Oxford

Zhang HM, Kim T (2001) A car-following theory for multiphase vehicular traffic flow. Presented at 80th TRB annual conference, Washington DC, 2001

Zheng P (2002) A microscopic simulation model of merging operation at motorway on-ramps. PhD thesis, University of Southampton, UK

Chapter 9
Traffic Simulation with Dynameq

Michael Mahut and Michael Florian

9.1 Model Building Principles

9.1.1 Introduction

Dynameq, which stands for "*dynam*ic *eq*uilibrium," is a simulation-based dynamic traffic assignment (DTA) model. The computational model consists of two main components: a traffic flow simulation model and a routing model. These two modules are concerned with different aspects of driver behavior. The routing model imitates how drivers choose their routes through the network to their desired destinations. The traffic flow simulation concerns all other aspects of the driving process: decisions to accelerate and decelerate due to traffic lights, signage and interactions with other vehicles, and the process of selecting a lane and executing a lane-change maneuver. The overall structure of the model is depicted in Fig. 8.1. As with all equilibrium approaches to the traffic assignment problem, the solution is an iterative method that repeats the simulation and routing computations many times over until it converges to a satisfactory solution. This procedure is analogous to the learning process of drivers in the real world repeating the same trips, such as the morning or afternoon commute, over a sequence of days.

At the start of each iteration (or "day"), the routing model generates the time-dependent path input flows, based on the time-dependant path travel times generated by the traffic simulation on the previous iteration (or "day"). The traffic simulation, more generically referred to as a *network loading* model, loads the network by simulating the movements of individual vehicles, as defined by the path input flows, as they make their journeys through the network. Thus, the outputs of the

M. Mahut (✉)
INRO Consultants Inc., 5160 Decarie Blvd. Suite 610, Montreal, QC, H3X 2H9, Canada
e-mail: michaelm@inro.ca

M. Florian
University of Montreal, C.P. 6128, Succursale Centre-ville, Montreal, QC, H3C 3J7, Canada and
INRO Consultants, Inc., 5160 Decarie Blvd. Suite 610, Montreal, QC, H3X 2H9, Canada
e-mail: mike@inro.ca

J. Barceló (ed.), *Fundamentals of Traffic Simulation*,
International Series in Operations Research & Management Science 145,
DOI 10.1007/978-1-4419-6142-6_9,

routing model are the inputs to the traffic simulation, and vice versa. The simulation model simultaneously generates various measures that describe the evolution of traffic flows through the network, such as flow rates, speeds, and densities for individual links, lanes, turns, and nodes. On the first iteration, in the absence of a previous iteration to generate link travel times, the *free-flow* travel times are used as inputs to the routing model.

The simulation model is based on the efficient discrete-event (event-based) traffic flow simulation model of Mahut (2001). The model is not as detailed as conventional discrete-time (time-step) simulation models (*microsimulation* models), but is nevertheless based on the same underlying sub-models, namely car following, lane-changing, and gap acceptance. The underlying design principle of the traffic flow simulation model in Dynameq is to provide an efficient trade-off between traffic flow fidelity (realism) and computation time. The low computation (CPU) times are particularly useful due to the iterative nature of the algorithm (see Fig. 8.1), which requires repeating the simulation many times over. The traffic flow simulation model is presented in detail in Section 9.2, *Core Traffic Flow Models*.

Mathematically, the DTA model is formulated as a time-discrete variational inequality and two solution methods are available. One is based on a straightforward adaptation of the method of successive averages (MSA) and the other on a heuristic adaptation of a gradient-based method used in solving the static network equilibrium model in the space of path flows. These methods can be considered to be heuristic since the dependence of the travel times on the link flows is complex and not given by an analytical function. This is due to the complexity of the traffic simulation which carries out the network loading step in the algorithm. A realistic representation of the system requires that the network loading properly represent traffic delays, i.e., in a way that is consistent with traffic flow theory. The resulting assignment map is discontinuous and difficult to characterize analytically.

The time-discrete nature of the assignment model means that the time-dependent path input flows are defined over a sequence of short time intervals, during each of which the probability of any given path being used for a given origin–destination (O–D) pair remains constant. These time intervals are referred to as *assignment intervals*, or sometimes simply *departure-time windows* (or *intervals*).

The routing model in this approach functions simultaneously as the route-generation model. A maximal number of required paths (N) is provided exogenously, and at each of the first N iterations, a time-dependent shortest path (TDSP) algorithm is used to determine the shortest path for each O–D pair and each departure-time interval. This path is added to the existing path set before the route input flows are re-calculated, thus gradually building up the set of paths and simultaneously dispersing the traffic over a wider set of paths with each iteration. After iteration N, the path set generally remains fixed. The iterations continue until a stopping criterion is satisfied, indicating that the current assignment is sufficiently close to dynamic equilibrium conditions. The assignment methods and stopping criteria are presented in detail in Section 9.3, *Dynamic Traffic Assignment*.

9.1.2 Model Building Principles: Dynamic Traffic Assignment

The traffic assignment model in Dynameq is a pre-trip dynamic equilibrium model. "Pre-trip" refers to the fact that each simulated driver makes a single path choice before departing on his trip, and this path is followed to the destination without being reconsidered on route. "Equilibrium" refers to the fact that the path choices, or path demands (in vehicles or vehicles per hour) in the solution of the model result in path travel times that approximately satisfy dynamic user-equilibrium conditions. These conditions are a time-varying extension of the Wardrop (1952) user-equilibrium conditions for static assignment: for any given departure time, a driver cannot improve his travel time by unilaterally changing paths. Pre-trip equilibrium assignment models are appropriate for off-line planning applications, which can range from short-term operational planning (e.g., impacts of road maintenance projects) to long-term travel forecasting exercises.

Friesz et al. (1993) formulated a dynamic equilibrium assignment model as an infinite dimensions variational inequality. The infinite dimension of the model is due to the fact that time is considered to be continuous. It is usual to consider a time-discrete formulation of the model, where time is subdivided into discrete intervals. Each interval is considered to be an interval for the departure of trips. The solution of the time-discrete formulation of the equilibrium dynamic traffic assignment problem seeks to obtain, for any given departure-time window, flows that equalize the travel times for all used paths for every O–D pair.

The extension of the Wardop user-equilibrium principle to the dynamic (time varying) context is based on *experienced* travel time, rather than *instantaneous* travel time. Instantaneous travel time implies that the path travel time is evaluated by adding up the link travel times (which are time-varying in a dynamic model) for the links of the path based on their values at a given *instant* in time. Using this definition, a given path, for the duration of a single trip along that path, has many different travel times, depending on when the travel time is evaluated. For example, microscopic traffic simulation models typically use instantaneous travel times since the demand is assigned to paths during a single execution of the simulation model (*one-pass assignment*). By contrast, a path has only one experienced travel time for any given trip, which is an estimate of the average travel time actually experienced by a driver, for a given path and departure-time window. Using the experienced travel time is also more behaviorally sound, for obvious reasons, when modeling habitual trips such as those during the morning and evening peak periods.

Since the experienced route travel times result from the interactions of the vehicles as they move through the network from their origins to their destinations, they cannot be known in advance, i.e., when the route choices are made. The path travel times are thus an input to the route decisions and an output, and this kind of cyclical problem can only be solved properly with an iterative approach such as shown in Fig. 9.1. As mentioned above, the iterations of the model can be thought of as a sequence of days over which drivers are adapting their route choices: on each day, before commencing the trip, the route choices are reconsidered based on the travel times experienced on the previous day.

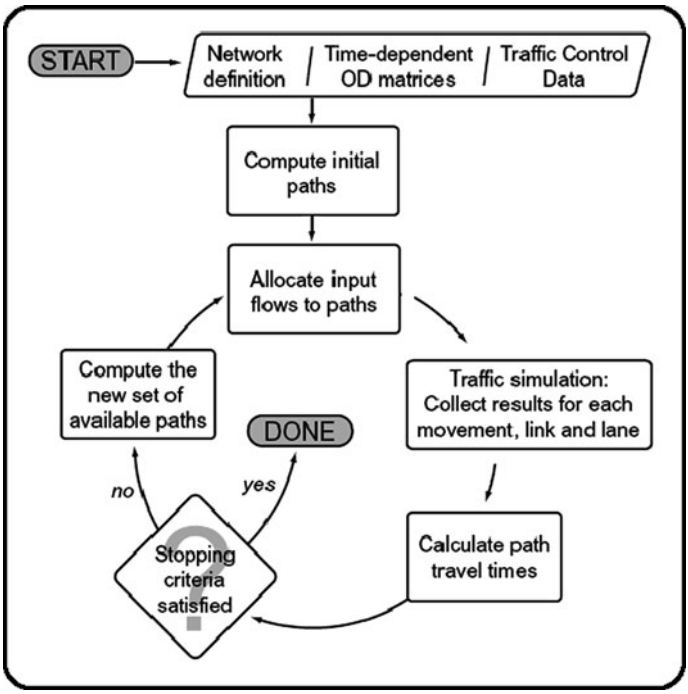


Fig. 9.1 General structure of solution algorithm

This gives the iterative solution a *predictive* property, because

(a) the routing decisions are based on an estimate of what traffic conditions *will be* along the route using the travel times of the previous iteration and
(b) as the model converges to a solution, the link travel times change relatively little from one iteration to the next.

This predictive property cannot be captured using *instantaneous* path travel time, since the instantaneous travel time measure is always based on what the link travel times are at the time of the decision, e.g., at the trip departure time.

This is true even if the instantaneous travel time is re-evaluated several times during the trip and the driver is allowed to re-consider the route at intermediate points from which an alternative route to the destination is available. For this reason, models based on instantaneous travel time, and in particular those which do not employ an iterative algorithm (i.e., only run the simulation once), are referred to as *reactive* models, because drivers make their route choices progressively, in response to the evolving traffic congestion on the network.

In many situations, particularly congested ones, the reactive approach can yield a significantly different solution from the equilibrium solution, i.e., obtained with an iterative (predictive) approach. Some DTA models use a *hybrid predictive/reactive*

approach, allowing drivers to react en route within an iterative solution method. In general including any reactive component to the route choice decision can increase the instabilities of the model solution and increase the probability of deadlock (gridlock) occurring in congested conditions. For this reason, reactive en route path switching is not currently modeled in Dynameq.

9.1.3 Modeling Building Principles: Traffic Flow Simulation

The traffic flow simulation model in Dynameq moves individual vehicles on a detailed (lane-based) network using car-following, lane-changing, and gap acceptance models. This type of traffic model is commonly referred to as a microscopic traffic simulation (or *micro-simulation*). From a practical standpoint, a microscopic traffic flow simulator can be defined as a model which explicitly represents the movements and interactions of individual vehicles, and in which the primary outputs, such a link flows, travel times, and densities, are a direct result of these interactions. By contrast, *macroscopic* models represent traffic as a fluid and are based on hydrodynamic or gas-kinetic descriptions of traffic flow (Hoogendoorn and Bovy, 1999; Diakaki and Papageorgiou, 1996; Messmer, 2000a; Messmer, 2000b; Papageorgiou, 1990; Richards, 1956; Lighthill and Whitham, 1955). It should be mentioned that in recent years, *macroscopic* has been used by practitioners to refer to static assignment models: the above definition is the traditional one from the traffic flow theory literature, and since it has no alternative names, this definition is maintained here, as is the term *static assignment model*.

A unique feature of the traffic simulation model in Dynameq is that it is solved using an event-based (*discrete-event*) algorithm, rather than the *time-step* (*discrete-time*) method typically employed in other traffic simulation packages, both commercial and academic (http://www.tss-bcn.com (Aimsun), accessed 12 Sep 2009; http://www.ptv.de (Vissim), accessed 12 Sep 2009; http://www.sias.com (Paramics), accessed 12 Sep 2009; http://sumo.sourceforge.net/ (SUMO), accessed 12 Sep 2009; http://web.mit.edu/its/products.html (MITSIMLab), accessed 12 Sep 2009; http://www.its.leeds.ac.uk/software/dracula (Dracula), accessed 12 Sep 2009; http://ops.fhwa.dot.gov/trafficanalysistools/corsim.htm (Corsim), accessed 12 Sep 2009; Van Aerde, 1999). Time-step and event-based models are fundamentally different approaches due to how they handle time. In a time-step model time is the independent variable, while in an event-based model, time is a dependent variable. These two paradigms are the primary approaches to building simulation models in general.

The main advantage of an event-based approach is that it can be much more computationally efficient than a time-step model, and this is the primary motivation for adopting an event-based simulator in Dynameq. However, it is usually more challenging to build an efficient event-based model than a time-step model, particularly for complex systems, because it is critical to design the event-based model in such a way as to minimize the number of events. The use of relatively simple car-following

and lane-changing models in Dynameq is directly tied to the fact that the model is solved with an event-based approach.

In recent years, the term *mesoscopic* has become quite widely used and generally refers to any model that falls somewhere between the macroscopic and microscopic definitions. Mesoscopic models arose from efforts to bring added realism to the macroscopic modeling paradigm. In practice, the term mesoscopic has also become somewhat synonymous with dynamic traffic assignment (DTA), since most DTA models employ some type of mesoscopic traffic modeling approach (Ben-Akiva et al., 1998; Ziliaskopulos and Lee, 1997; Mahmassani et al., 2001; Leonard et al., 1989). Dynameq's microscopic approach, although embedded within an iterative DTA model, is fundamentally different from this type of mesoscopic traffic flow model.

In effect, what all DTA models have in common is the type of problem they are trying to solve, more so than the specific ways in which they solve it. The term mesoscopic applies equally well to aspects of the problem, such as the typical network size – which falls somewhere between those that are commonly handled by microscopic simulators and static assignment models – and the required level of detail, or realism, of the modeled system.

The appropriate level of detail is determined by two competing factors:

- the system being modeled has increasingly complex and sophisticated elements, from adaptive traffic signal control to variable message signs to high-occupancy toll (HOT) lanes, which require a relatively high-fidelity model to be properly represented and evaluated;
- the scale of the network makes it impractical to have an exact, complete, and error-free representation of the physical system in the model;

The need to reconcile these two considerations is the main challenge in building a DTA model. Specifically, there is a need to avoid *false degrees of precision* by maintaining consistency between the level of detail (complexity) of the model components and the known precision of the input data. Moreover, there needs to be consistency in the precision that is assumed by the different components of the model, such as route choice, lane-selection, gap acceptance, and car following. For example,

- it matters little if a car is braking at 2.0 or 2.5 m/s^2 if the car really should be in a different lane;
- similarly, it matters little if the car is in the correct lane if it is not on the right route;
- and ultimately, being on the right route is unimportant if the origin or destination of the trip are not correct, i.e., if the trip should not even have been made (modeled).

Understanding the relative importance of the various model components and their related input data, and seeking consistency among them, is what ultimately leads to

an efficient model implementation that provides an optimal trade-off between the quality and usefulness of the results, and the computational burden and human effort required for collecting the input data and calibrating an application of the model.

9.2 Core Traffic Flow Models

As with all traffic simulators, the core of the model is the underlying car-following model and its solution method. The traffic flow simulator in Dynameq is based on the following simplified car-following model:

$$x_f(t) = \min\left[x_f(t-\varepsilon) + \varepsilon V, x_l(t-R) - L\right] \tag{9.1}$$

where $x(t)$ is the trajectory of a vehicle (position as a function of time), L is the effective vehicle length, R is the driver/vehicle response time, V is the free-flow speed, and ϵ is an arbitrarily short time interval. The subscripts f and l denote the trajectories of two vehicles in sequence, one following and the other leading, respectively. The first term inside the min operator represents the farthest position downstream that can be attained at time t based on the follower's position at time $t - \varepsilon$, as constrained by the maximum speed of the vehicle, V. The second term inside the min operator represents the farthest position downstream that can be attained based on the trajectory of the next vehicle downstream in the same lane, using a simple collision-avoidance rule (Mahut, 2001; Mahut, 1999; Newell, 2002).

This car-following model is referred to as a simplified car-following model, or lower order model, since it only defines the position of each vehicle in time, rather than vehicle speed or acceleration. Traditionally, and most commonly, car-following models define the acceleration $a_f(t)$ as a function of the state variables of the follower and the leader at time $(t - R)$ (Brackstone and McDonald, 1999; Gabard, 1991). When these models are solved, i.e., using a discrete-time approach, the trajectory of each vehicle is characterized by constant acceleration over short time intervals, from which vehicle speed and position can be computed (with appropriate boundary conditions). In the simplified model used here, the trajectory is characterized by constant speed over short time intervals. In this form, the model can also be seen as a continuous-time analogy to cellular automata models used for traffic simulation (Nagel and Schreckenberg, 1992).

This model is solved using an event-based solution, which first requires converting the statement of the car-following relationship in eq. (9.1) from $x(t)$ to $t(x)$, which yields

$$t_f(x) = \max\left[t_f(x-\delta) + \frac{\delta}{V}, t_l(x+L) + R\right] \tag{9.2}$$

From this relationship one can derive the following expression, which only calculates the link entrance and exit time of each vehicle:

$$t_n(X_1) = \max\left[t_n(0) + \frac{X_1}{V_1}, t_{n-1}(X_1) + R + \frac{L}{\min[V_1, V_2]}, t_{n-X_2/L}(X_2) + \frac{X_2}{L}R\right] \tag{9.3}$$

where X_1 and X_2 are the lengths of two sequential links, with speeds V_1 and V_2, respectively. The subscript n indicates vehicle numbering in sequential order, i.e., vehicles n and $n-1$ represent a follower and leader, respectively. The vehicle attributes represented by L and R are assumed to be identical over the entire traffic stream, and each vehicle adopts the link-specific free-flow speed when traversing a given link. The link lengths are assumed to be integer multiples of the vehicle length L.

This "link-based" solution (9.3) provides a very practical and computationally efficient way to model traffic on a single lane, i.e., to rigorously solve the car-following model (eq. (9.1)) over a linear sequence of links without actually calculating the position of each vehicle at each second or less (using a time-step solution). The ability to rigorously calculate longitudinal traffic dynamics over entire links has also been demonstrated for the kinematic wave model based on the two-segment linear (triangular) relationship between traffic flow and density (Newell, 1993), often called the fundamental diagram. Not surprisingly, the three-parameter car-following model shown here (eq. (9.1)) also yields the triangular flow–density relationship (Mahut, 2001; Mahut, 1999).

A multi-lane version of the above relationship (eq. (9.3)) maintains the same property of only calculating the entrance and exit times of each vehicle and also captures the interactions between vehicles due to lane-changing maneuvers (Mahut, 2001). This multi-lane extension requires each driver to select his departure lane upon entering a link and computes the resulting delay effect of a single lane-change maneuver – across several lanes, if necessary – which occurs at the first position on the link at which the vehicle encounters a delay propagating from downstream, on any of lanes spanned by the maneuver. The intent of the multi-lane model is to capture the reductions in effective (operational) capacity on links, such as freeway segments, where a significant amount of lane changing occurs, particularly due to mandatory lane changes that drivers must execute in order to remain on their intended paths. The model also employs a complex set of heuristics for modeling a driver's lane-selection decisions (Florian et al., 2008). It has also been extended to allow vehicle length and driver response time to vary individually by vehicle (Florian et al., 2008). The model can thus be characterized as a continuous-space, continuous-time, discrete-flow (vehicle-based) model that employs a lane-based representation of the network.

The above solution (eq. (9.3)), or the multi-lane version of it (Mahut, 2001), provides the time at which a vehicle crosses the node between two sequential links, where the node in question joins only those two links. However, this relationship can be easily extended to handling nodes with multiple incoming and outgoing links, considering conflicts between vehicle trajectories and including an explicit representation of traffic signals. This primarily requires including an additional term inside the *max* operator in (eq. (9.3)) to include constraints based on conflicting

vehicles that have crossed the node prior to the vehicle in question, and then applying this formula to all vehicles waiting to cross a given node in the network. A gap acceptance model is then applied to resolve conflicts between these vehicles as required (discussed below), and the next vehicle to cross the node is determined. This process allows the traffic flow component of the simulation model to be solved with an event list that is of the same size as the number of nodes in the network. A more complete description of the event-based algorithm is given in Mahut (2001).

For modeling gap acceptance behavior, i.e., the interaction between two vehicles with conflicting trajectories (e.g., at an uncontrolled intersection), Dynameq employs a two-parameter gap acceptance model. The decision of whether a lower priority vehicle will precede a higher priority vehicle at the conflict point is based on two quantities, which are as follows:

- the *available gap* (g): the time difference between the arrival of the two vehicles to the conflict point (higher priority vehicle minus lower priority vehicle);
- the *relative waiting time* (w): the difference between the time spent waiting at the stop line for an available gap (lower priority vehicle minus higher priority vehicle);

These two quantities are used in conjunction with the following two parameters:

- the *critical gap* (G): the value of *available gap* at which there is a 50% chance of the lower priority vehicle preceding the higher priority vehicle;
- the *critical wait* (W): the value of *relative waiting time* at which there is a 50% chance of the lower priority vehicle preceding the higher priority vehicle;

The probability that the lower priority vehicle precedes the higher priority vehicle, called the *precedence probability* (P), is then computed as follows:

$$P = \min\left[\max\left[\frac{g}{G} - 0.5, \frac{w}{W} - 0.5, 0.0\right], 1.0\right] \tag{9.4}$$

This model considers the effects of available gap and waiting time independently by taking the maximum of two linear density functions. Each function increases from zero to unity over the range ($x/2$, $3x/2$), where x represents a model parameter (G or W). The waiting time term takes into account the effect of a driver's impatience when he is unable to enter a conflicting traffic stream. Practically speaking, it ensures a minimum flow rate on the lower priority turning movement under heavily congested conditions in which no gaps of reasonable size are available.

As described above, the simulation model is solved using an event-based algorithm, which is a fundamentally different approach from time-step models. In a time-step model, the state variables of all vehicles are updated at the end of each discrete-time interval (typically between 0.1 and 1.0 s), based only on the known state variables at the previous time-step. In addition to the potential efficiency in

computation that can be achieved with an event-based approach – albeit with a reduction in the complexity of the car-following and lane-changing models – some other basic differences between event-based and time-step models should be briefly mentioned.

The results generated by a time-step model will depend on the selected size of the time-step, and thus changing the time-step will inevitably change the results. If the time-step is small enough, then making it smaller should not have a significant impact on the results, but how small is *small enough* depends on the model and is not always easy to determine. The conventional approach is to make the time-step equal to the driver response time, which as a result has to be common across all vehicles (for smaller time-steps, all driver response times must be multiples of the time-step used). The appropriate size of the time-step, in order to properly apply a gap acceptance model and avoid undesirable model properties, is an ongoing topic of research and discussion (Chevallier and Leclerq, 2009).

Event-based models, by contrast, do not employ a time-step and thus produce a single set of results for any given set of inputs (and for a given random seed, of course, as all microsimulators include stochastic components which require the use of quasi-random number generation). Specifically, the lack of a time-step in an event-based model ensures that for the given inputs, the *correctness* of the outputs is not subject to the selection of an appropriate time-step size. Moreover, driver response time can be real-valued and drawn randomly from an appropriate distribution. This allows computations that involve quantities on the order of the response time to be solved rigorously and to properly capture the impact of the assumed response time distribution on gap acceptance behavior.

9.3 Dynamic Traffic Assignment

9.3.1 Mathematical Model

In this section the time-discrete formulation of the equilibrium dynamic traffic assignment model is stated.

The path choices are modeled as decision variables governed by a user-optimal principle where each driver seeks to minimize his experienced path travel time. All drivers have perfect access to information, which consists of the travel times on all paths (used and unused). The solution algorithm takes the form of an iterative procedure designed to converge to these conditions.

The solution approach adopted for solving the dynamic network equilibrium model, eqs. (9.5), (9.6), and (9.7), is based on a temporal discretization into periods $\tau = 1, 2, ..., \left|\frac{T_d}{\Delta t}\right|$, where Δt is the chosen duration of a departure-time interval. This results in a time-discrete model.

The mathematical statement of a time-discrete version of the dynamic equilibrium problem is in the space of path flows h_k^τ, for all paths k belonging to the set K_i for an origin–destination $i \in I$, at departure time t. The time-varying demands are

denoted $g_i^\tau,\ i \in I,\ all\ \tau$. The path flow rates in the feasible region Ω satisfy the conservation of flow and non-negativity constraints

$$\Omega^\tau = \left\{ h_k^\tau : \sum_{k \in K_i} h_k^\tau = g_i^\tau,\ i \in I,\ all\ \tau\ ; h_k^\tau \geq 0,\ k_i,\ i \in I,\ all\ \tau \right\} \tag{9.5}$$

and a temporal version of Wardrop's (Wardrop, 1952) user-optimal route choice results in the model:

$$h_k^\tau \in \Omega,\ u_i^\tau(t) = \min_{k \in K_i} \left\{ s_k^\tau(t) \right\} \tag{9.6}$$

$$s_k^\tau \begin{cases} = u_i^\tau\ if\ h_k^\tau > 0 \\ \geq u_i^\tau\ if\ h_k^\tau = 0 \end{cases} \text{for all } k \in k_i,\ i \in I,\ \tau = 1, 2, ..., \left| \frac{T_d}{\Delta t} \right| \tag{9.7}$$

which can be shown to be equivalent to solving the discrete variational inequality (Friesz et al., 1993).

$$\sum_\tau \sum_{k \in K} s_k^\tau(h^{\tau *})\ (h_k^\tau - h_k^{\tau *}) \geq 0 \tag{9.8}$$

where $K = \bigcup_{i \in I} k_i$ where h^τ is the vector of path flows (h_k^τ) for all k and τ.

The demonstration of existence and uniqueness of a solution to this model depends on the dependence of link and path travel times on the path input flows and the dependence of the path input flows on the link and path travel times. Since the properties of these mappings are not easily verified, due to the fact that it is the output of a simulation model and not an analytical transformation, no claims are made about the existence or the uniqueness of a solution. The equilibrium principle is simply used as a guide to compute an approximate solution of the time-discrete variational inequality.

The next sections present an MSA-based solution algorithm to this problem, followed by an algorithm inspired by the projected gradient method. A heuristic method which allows the maximum step size to increase with departure time, which is applicable to both the MSA and the gradient-like algorithms, is presented afterward.

9.3.2 MSA-Based Algorithm

As mentioned above, and shown in Fig. 9.1, the solution algorithm consists of two main components other than the computation of the temporal shortest paths: a method to determine a new set of time-dependent path input flows, based on the experienced path travel times of the previous iteration, and a method to determine

the actual link flows and travel times that result from a set of path inflow rates. The algorithm furthermore requires a set of initial path flows.

The path input flows $h_k^\tau,\ k \in K$ are determined by a variant of the method of successive averages (MSA), which is applied to each O–D pair i and time interval τ. An initial feasible solution is computed by assigning the demand for each time period to a set of successive shortest paths. Starting at the second iteration, and up to a pre-specified maximum number of iterations, N, the time-dependent link travel times after each loading are used to determine a new set of dynamic shortest paths that are added to the current set of paths.

For all iterations $l, l \leq N$, the volume assigned as input flow to each path in the set is $\frac{g_i^\tau}{l}, i \in I,\ all\ \tau$. Subsequently, for iterations $l, l > N$, only the shortest among *used* paths is identified and the path input flow rates are redistributed over the known paths as described below.

If the flow of a particular path decreases below a small predetermined value then the path is dropped and its remaining flow is distributed proportionally to the other used paths. This heuristic approach is akin to the restricted simplicial decomposition algorithm of Lawphongpanich and Hearn (1984) for the solution of the static network equilibrium model with fixed demand. The stopping criteria are the maximum number of iterations, L, and a maximum average relative gap, denoted γ. The relative gap measure is discussed below, after the statement of the algorithm.

MSA Equilibrium DTA Algorithm

- *Step 0 Initialization (iteration counter $l = 1$):*
  Compute temporal shortest paths based on free-flow travel times.
  Load the demands (traffic simulation) to obtain an initial solution;
  Update iteration counter: $l = l + 1$.
- *Step 1 Reallocation of input flows to paths:*
  Step 1.1 If ($l \leq N$)
  Compute a new dynamic shortest path.
  Assign to each path k the input flow

$$h_k^{\tau,l} = \frac{g_i^\tau}{l}, i = 1, 2, , , |I| \tag{9.9}$$

  Step 1.2 If ($l > N$)
  Identify the shortest among used paths.
  Redistribute the flows as follows:

$$h_k^{\tau,l} = \begin{cases} h_k^{\tau,l-1}\left(\frac{l-1}{l}\right) + \frac{g_i^\tau}{l}, & \text{if } s_k^{\tau,l} = u_k^{\tau,l}; k \in K_i, i \in I, \text{ all } \tau \\ h_k^{\tau,l-1}\left(\frac{l-1}{l}\right) & \text{otherwise} \end{cases} \tag{9.10}$$

- *Step 2 Stopping rule:*

  If $l \leq L$ *or* $RGap \leq \gamma \Rightarrow STOP$;
  Otherwise, return to Step 1

While no formal convergence proof can be given for this algorithm, since the network loading map does not have an analytical form, a measure of gap, inspired from that used in static network equilibrium models, may be used for qualifying a given solution. It is the difference between the total travel time experienced and the total travel time that would have been experienced if all vehicles had the travel time equal to that of the current shortest path (for each interval τ).

Hence a relative gap for each departure-time interval τ and iteration l may be computed as

$$R\ \mathrm{Gap}^{\tau,l} = \frac{\sum_{i\in I}\sum_{k\in k_i} h_k^{\tau,l} s_k^{\tau,l} - \sum_{i\in I} g_i^{\tau,l}\ u_i^{\tau,l}}{\sum_{i\in I} g_i^{\tau,l}\ u_i^{\tau,l}} \tag{9.11}$$

where $u_i^{\tau,l}$ are the lengths of the shortest paths at iteration l. A relative gap of zero would indicate a perfect dynamic user-equilibrium flow. Clearly this is a fleeting goal to aim for with any simulation-based dynamic traffic assignment.

It is very important to note that this model, even though its general formulation is very similar to flow-based models, is in fact a simulation model that moves individual cars on the links of the network, as discussed in Section 9.2, *Core Traffic Flow Models*.

9.3.3 Gradient-Like Algorithm

The equilibration algorithms used in static equilibrium models that operate in the space of path flows provide some ideas that may be adapted heuristically for the solution of the dynamic equilibrium traffic assignment problem. These algorithms are adaptations of the classical convex simplex, projected gradient and reduced gradient algorithms implemented with a Jacobi or a Gauss Seidel decomposition scheme. Some selected references on the topic are (Dafermos, 1971; Leventhal et al., 1973; Patriksson, 1994).

In particular, it is very attractive to adapt the equivalent of the projected gradient algorithm, even though there is no formal objective function that can be identified and the model formulation is a time-discrete variational inequality. Since there is no objective function the step sizes used are those of the MSA method (or the modified MSA method described below) and are adapted to ensure that the path flows remain non-negative.

Before stating the mathematical model it is useful to review the general steps of the adaptation of the projected gradient method (Rosen, 1960; Luenberger, 1984) for the static network equilibrium problem. For each O–D pair the general steps of this algorithm, stated qualitatively, are the following:

1. compute the average cost of all used paths (by O–D pair);
2. reduce the flow of paths that have a larger cost than the average;

3. increase the flow on paths that have a smaller cost than the average;
4. only keep the paths with positive flow;
5. add a path if it is shorter than the current equilibrated solution.

The same basic idea is adapted for the equilibrium dynamic traffic assignment algorithm presented here. The only difference is that the method is used as a heuristic and is applied to each departure interval. In the static model, the step size is computed by minimizing the objective function over the paths that change flow. For dynamic assignment, the default step size is the MSA step size. However, it must also be constrained by the smallest step size that annuls the flow on any path (for a given O–D pair and departure-time interval).

In order to state the algorithm (for one O–D pair) the notation used is the following. Let K^+ be the set of paths with positive flow. Let s_k be the cost (time) of a path, and $\bar{s}$ be the average value of the path costs; p_k is the proportion of input flow to the path $k \in K^+$, that is, $p_k = h_k / g_i$; d_k is the direction of change for each path and d_k^n is the normalized direction; α_{MSA} is the MSA step size.

The gradient-like algorithm modifies the flow changes by using the following steps:

Compute the vector of $d_k = \bar{s} - s_k$, $k \in K^+$;

The "direction" vector d_k^n is normalized, $d_k^n = \frac{\bar{s} - s_k}{\sum_k |d_k|}$, in order to satisfy conservation of flow conditions. The largest step size $\alpha_{\max}$ which would diminish the input proportion of a path to zero, is $\alpha_{\max} = \max\left[\frac{p_k}{d_k^n} \middle| d_k^n < 0\right]$. The largest actual step size is then $\alpha = \min(\alpha_{\text{MSA}}, \alpha_{\max})$ which is used to update the path proportions $p_k = p_k + \alpha d_k^n$ and the new path input flows are $g \cdot p_k$.

Next the DTA algorithm is stated for the gradient-like algorithm based on the adaptation of the projected gradient steps.

Gradient-Like Equilibrium DTA Algorithm

- *Step 0 Initialization (iteration counter $l = 1$):*
  Compute temporal shortest paths based on free-flow travel times.
  Load the demands (traffic simulation) to obtain an initial solution;
  Update iteration counter: $l = l + 1$.

- *Step 1 Reallocation of input flows to paths:*
  Step 1.1 If ($l \leq N$)
  Compute a new dynamic shortest path.
  Assign to each path k the input flow $h_k^{\tau,l} = \frac{g_i^\tau}{l}$
  Step 1.2 If ($l > N$)
  Compute the vector of $d_k = \bar{s} - s_k$,

$$k \in K^+ \tag{9.12}$$

  Normalize the vector

$$d_k^n = \frac{\bar{s} - s_k}{\sum_k |d_k|}$$

Check for $\alpha_{\max}$, the largest value of α :

$$\alpha_{\max} = \max\left[\frac{p_k}{d_k^n} \,|\, d_k^n < 0\right]; \alpha = \min\left(\alpha_{\text{MSA}},\ \alpha_{\max}\right) \tag{9.14}$$

Update the path proportions $p_k = p_k + \alpha d_k^n$
Redistribute the flows as follows:

$$h_k^{\tau,l} = g_i^{\tau *} p_k^{\tau,l}; k \in K_i, i \in I,\ all\ \tau \tag{9.15}$$

- *Step 2 Stopping rule:*
  If $l \leq L$ *or* $RGap \leq \gamma \Rightarrow STOP$;
  Otherwise return to Step 1

Once the path proportions are computed (with either algorithm), they are used by the *vehicle generation* process of the simulation model to generate discrete vehicles with individual (random) departure times. This is a stochastic process which, on average, will produce a number of vehicles for each path that corresponds to the product of the theoretical path input flow $\left(h_k^{\tau,l}\right)$, which has units vehicles/h, and the duration of the departure-time interval. Clearly, the simulation model must load discrete vehicles onto the network, while the above product is real-valued. This is handled in the standard way by interpreting the theoretical path proportions (p_k) as path probabilities. If the O–D demand for each time interval is not defined as an integer number of vehicles, a standard matrix rounding technique is employed to convert the real-valued matrix to integers.

9.3.4 Time-varying Step Size Adjustment

For the vast majority of the real-world applications of this model to date, the relationship between departure time and relative gap (after the algorithm stops) is monotonically non-decreasing, i.e., the assignment for a departure-time interval is further away from the equilibrium conditions than for the preceding interval. Another consistent trend is that later departure-time intervals require more iterations before converging to a stable value of relative gap. These observations are presented in more detail in Section 9.4, *Calibration and Advanced Modeling Features*.

One explanation for these phenomena is that the travel times of later-departing vehicles are affected by earlier-departing vehicles, and thus the convergence for a later-departing interval cannot be achieved until it has first been achieved for the prior interval. This inherent property of the model suggested the possibility that

the higher values of relative gap in the later-departing intervals might be partially a result of the fact that the MSA step size is the same for all departure-time intervals at each iteration. To put it simply, by the time (in iterations) that a later interval finally starts to converge, the step size is so small that not enough flow is being moved away from the longer paths toward the shorter paths. Another reason for the increasing values of relative gap is that later-departing vehicles incur higher congestion. A positive correlation between congestion levels and relative gap (after the algorithm stops) has also been consistently observed across various networks.

These observations are the basis of a time-varying step size heuristic. The heuristic uses an integer *reset* parameter n and is first applied in step 1.2 of the algorithms presented above. The first N iterations, as described by step 1.1, remain unchanged. The modifications to step 1.2 are as follows. Let the number of departure-time intervals be $D = \left|\frac{T_d}{\Delta t}\right|$, and let $\tau = 0, 1, 2, ...D-1$ denote the departure-time intervals in increasing order. The first $n \cdot D$ iterations of step 1.2 are a transitory period during which the MSA step size, normally defined as $\alpha_{MSA} = \frac{1}{l}$ is modified by adjusting the iteration number in the denominator in a way that varies with the departure interval τ. The modified step size parameter, ${\alpha'}_{MSA}^{\tau,l}$, is calculated as follows:

$$ {\alpha'}_{MSA}^{\tau,l} = \begin{cases} \frac{1}{l-\left(\left\lfloor\frac{l-N}{n}\right\rfloor+1\right)n} & if\ \tau > \left\lfloor\frac{l-N}{n}\right\rfloor \\ \frac{1}{l-\tau\cdot n} & \text{otherwise} \end{cases} \tag{9.16} $$

This method calculates an index value $\left\lfloor\frac{l-N}{n}\right\rfloor$, which increments by one every n iterations. At each iteration where this value is incremented, the denominator is decremented by n for all departure intervals $\tau > \left\lfloor\frac{l-N}{n}\right\rfloor$, where $\tau = 0$ denotes the first departure interval. After iteration $l = N + n \cdot (D-1)$, the step sizes are simply

$$ {\alpha'}_{MSA}^{\tau,l} = \frac{1}{l-\tau\cdot n}, $$

That is, the inverse of the step size ${\alpha'}_{MSA}^{\tau,l}$ for each departure interval is n less than that of the previous interval. This pattern remains for all subsequent iterations until the algorithm stops. Figure 9.2 shows this heuristic rule in a visual way as a graph of ${\alpha'}_{MSA}^{\tau,l}$ vs. iteration number for $D = 6$, $N = 10$, and $n = 5$.

This heuristic, along with the gradient-like method, is presented in more detail along with numerical tests in Mahut et al. (2008).

9.4 Calibration and Advanced Modeling Features

This section provides information relevant to calibration of simulation-based DTA models, based on experience with applying Dynameq on real-world networks. The methodology is related to outputs, analysis tools, and model properties specific to the Dynameq traffic flow model and equilibration scheme.

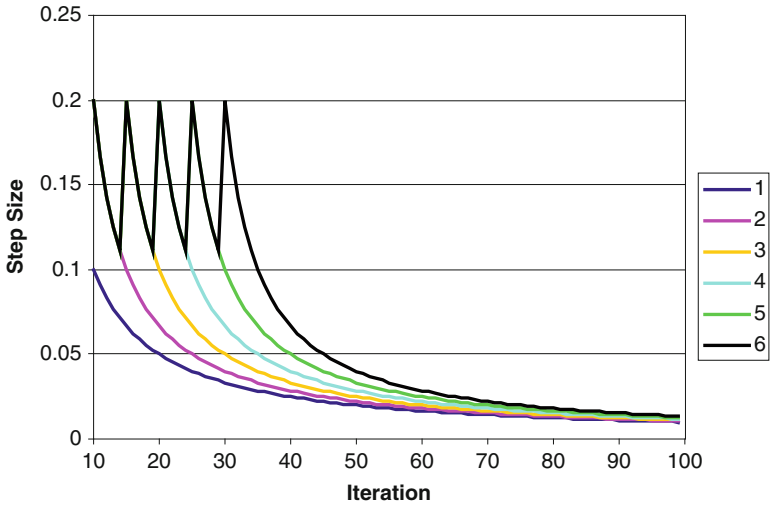


Fig. 9.2 Step size vs. iteration for six departure intervals

Calibration refers to the process of adjusting model inputs in order to improve the fit of the model outputs to field observations. Inputs can be justifiably modified in this context for a number of reasons:

– *For practical purposes, many model inputs are not explicitly measured in reality and are instead represented by default values*. These default values are adjusted, only as is found to be necessary, after comparing the model outputs to corresponding field observations. A typical example is the maximal flow rate (*ideal saturation flow rate*, or *capacity*) that can be sustained on a roadway which can vary somewhat even between roads of the same functional category. However, it can be reasonably well approximated, in most cases, with default values. On occasion, it may be necessary to adjust this value, or parameters known to directly affect it (if it is not an explicit input).
– *The inputs in question cannot be measured accurately and are thus only known with some degree of uncertainty*. A typical example is the travel demand data underlying an application, in the form of a time-varying origin–destination matrix.
– *Due to known limitations of the model, accurate inputs for the available parameters will not yield sufficiently accurate outputs in some instances.* There may often be rules of thumb for adjusting certain inputs in these situations. One example is the effective (operational) capacity of weaving sections on freeways, which can be relatively unstable and difficult to predict, even in reality, and thus difficult to model accurately. Another example is driver route choice behavior, which can follow subtle individual behavior that is difficult to represent directly with the model routing algorithms. For example, the model may make use of an off-ramp/on-ramp

sequence in an attempt to by-pass heavy congestion on a freeway, even though this behavior is not observed in the real world.

The actual process of calibrating a DTA model is dependent on the analysis tools that are available in the software package and requires a thorough understanding of the properties and characteristics of the model. For example, a model that is based on a microscopic traffic simulation, using a lane-based representation of the network, exhibits sensitivities to the inputs that are not captured using a less detailed approach. Thus the calibration process must be catered to the specific tool and take into consideration its strengths and limitations. The reader is referred to Chiu et al. (2010) for a good overview of DTA modeling concepts, including calibration and data-related issues.

Since the calibration approach is tied closely to the embedded analysis tools and software features in general, the discussion below includes some brief descriptions of advanced modeling features where relevant.

9.4.1 Calibration and Stability

The convergence measure associated with the solution of a DTA model – the *relative gap* measure mentioned above – provides critical information about how well the current assignment satisfies the equilibrium property, i.e., how equal the travel times are on alternative paths. A more general type of convergence measure (not used in Dynameq), which does not provide such information, is one which simply indicates how much the algorithm is actually changing the inputs (e.g., path demand flows) or outputs (e.g., link flows) from one iteration to the next. This kind of measure only gives an idea of how stable the current solution is, but does not quantify the solution with respect to the underlying equilibrium (user-optimal) objective. For this reason, such convergence measures can be deceiving: they indicate only that the algorithm is no longer improving the results, but does not indicate how well the final solution satisfies the desired objective of equilibrating travel times on alternative paths.

A typical plot of relative gaps against iteration number, for a sequence of departure-time windows, is shown in Fig. 9.3. The values of relative gap indicate how well the travel times on alternative paths are equilibrated, where zero would indicate a perfect equilibrium. A term that is sometimes used to refer to the property of an assignment (DTA solution) being in approximate equilibrium is *consistency.* This term is used in reference to the fact that at equilibrium, the path flows are *consistent* with the assumed behavioral mechanism of each driver attempting to minimize her own travel time (or generalized cost). The idea of consistency is exactly what is represented by convergence measures such as the relative gap. More general convergence measures, as mentioned in the previous paragraph, that do not quantify the solution in this way do not provide information about the consistency of the assignment.

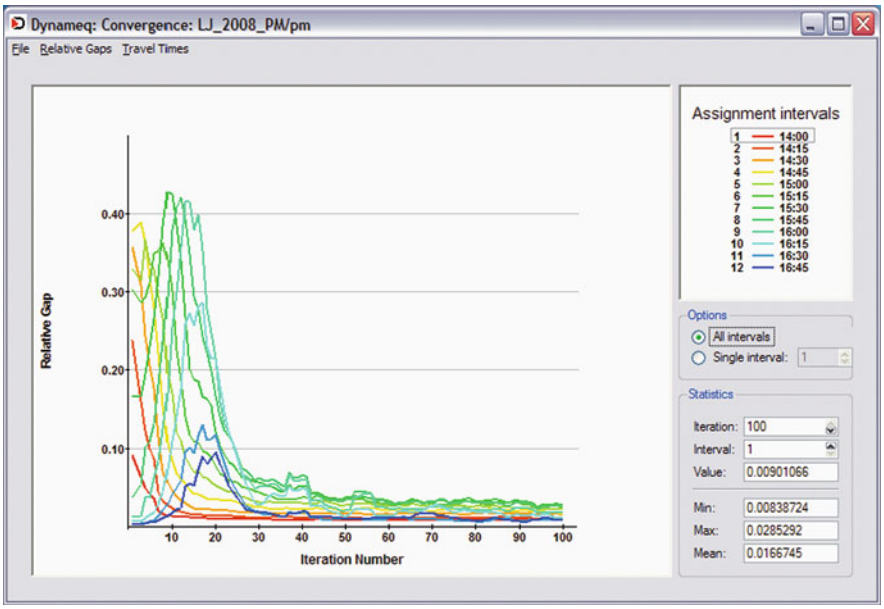


Fig. 9.3 Relative gaps for 12 departure intervals

The plot shows some basic properties that are common to most DTA applications:

- after a certain number of iterations, the relative gap for any given departure-time interval becomes relatively stable, after which it no longer appears to improve (i.e., does not tend to zero with increasing iterations);
- the relative gaps for a given departure-time interval begin to stabilize only after the previous interval begins to stabilize, in a sort of "domino effect";
- at any given iteration, and in particular in this stable region, the value of relative gap tends to increase with departure time, though in some cases (as in Fig. 9.3) the relative gaps will begin to decrease over the last few intervals;
- relative gap values, and in particular the stable values attained, tend to increase with increasing congestion in the network;
- the stable values of relative gap are generally some orders of magnitude higher than what is typically considered acceptable for static assignment models; this is due to the underlying cost dependence on flows which is highly nonlinear and discontinuous, as well as the discrete nature of the traffic representation.

In this context, typical applications are networks of less than 10,000 links, with demand periods of not more than 3 h (a typical AM or PM peak period model).

Since DTA models, i.e., the solution algorithms used for solving these models, do not converge to perfect equilibrium on networks of any significant size, the

practical stopping criterion for these models depends on identifying the stable solution. Generally, after reaching stable values of relative gap, the outputs of interest are no longer changing significantly from one iteration to the next. Moreover, the goodness-of-fit statistics (between model outputs and empirical data) should not change significantly if the model is stopped one iteration sooner or later. If this were not the case, i.e., if the DTA were not stopped at a stable solution or if the algorithm cannot produce a stable solution, the calibration exercise would be completely meaningless. As a general rule, smoothness in the relative gap plots prior to the last iteration strongly indicates this kind of stability.

The idea of stability is also used in conjunction with modeling in a rather different way, where it indicates how sensitive the (equilibrium) results are to small changes in the model inputs. For example, if closing one lane of traffic or changing the free-flow speed on a single network link drastically affects the congestion in the network, it may be said that the model results are somewhat unstable with regard to these inputs. This is a different notion of stability from that discussed above, in that it involves a comparison between two sets of DTA results, both of which are converged and stable (which is what allows them to be meaningfully compared in the first place) – but, each set of results is obtained from a slightly different version of the same network.

Although these two notions of stability seem to be quite disconnected, there is one important way in which they must be understood together. When comparing two models with slightly different inputs (e.g., when comparing alternative freeway improvement projects) significant differences in the outputs may be due to the fact that the models are not being run to equilibrium (i.e., to a *stable solution*), rather than because the (equilibrium) model outputs are really that sensitive to the physical differences between the two scenarios. In this case, instability of the first kind (the stability of a given DTA run) is being mistakenly attributed to an instability of the second kind, i.e., the results being sensitive to the differences in network topology between the two scenarios.

These concepts are particularly relevant to DTA modeling due to the prevalent use of *en route assignment* (or *en route path switching*) embedded in virtually all traffic simulation models, and in some DTA models as well. The most extreme case is the reactive *one-pass assignment*, discussed in Section 9.1.2, *Model Building Principles: Dynamic Traffic Assignment*, in which vehicles are constantly changing paths in response to evolving traffic conditions, but with no notion of where congestion will be encountered further downstream on the path (other than using current traffic conditions as a proxy). Such models may exhibit unstable behavior, in the sense that small changes to the inputs can result in larger than expected changes to the outputs. In this case, the instability is due to the fact that the one-pass approach does not necessarily provide a good approximation of equilibrium conditions, which is akin to an iterative model not being run for enough iterations. Perhaps because *one-pass* models do not generally provide a measure such a relative gap (though in principle they could) indicating how well the experienced path travel times (or generalized costs) were ultimately balanced out, the danger of instability in these models is often overlooked.

9.4.2 Calibration: Overview

The process of calibrating a DTA model can be broken down into two sequential analysis stages: qualitative and quantitative. The qualitative analysis stage is what typically starts after the very first model runs, when there may still be numerous errors in the input data to be found. In these situations it may be of little practical value to begin comparing the model outputs to empirical data, especially if the model is not converging to a stable solution. Once the model has been improved to a certain extent the quantitative analysis starts. Quantitative analysis is based on a direct comparison of model outputs and empirical data and investigating the outliers in order to further refine the model. In principle, fixing coding errors is not thought of as part of the calibration process; in practice, these two tasks are inseparable.

9.4.2.1 Qualitative Analysis

As discussed above, convergence measures indicate the quality of a DTA run, and should provide information about how well the path travel times (or generalized costs) are equilibrated. These measures are particularly important in the early stages of the calibration when they are most likely to indicate that the model results are unsatisfactory due to an unconverged or unstable solution.

As a general rule, errors in network and traffic signal coding are found to cause more congestion rather than less, due to the nature of congestion spillback: as a queue grows in space it engulfs vehicles that do not directly contribute to the original cause of the queue (their paths take them off the road before reaching the downstream bottleneck). In extreme cases, queues that are initially separate become connected as they grow, causing congestion to grow even faster and spread out in many directions, which can even lead to gridlock. Applications with gridlock typically exhibit unstable convergence, or even a complete failure to converge at all.

Under such circumstances, the DTA results are unsuitable for comparison to empirical data. There is a need to identify the key bottlenecks underlying the congestion and to correct the input errors that result in inaccurate capacity values, incorrect routing, or unrealistic demand. In general, the purpose of the qualitative analysis stage is primarily to achieve model results that exhibit a stable solution, are free of gridlock, and if possible, in which the overall congestion pattern at least resembles the observed conditions on the street.

A characteristic of this calibration stage is that correcting a single input value, e.g., adding a missing turn pocket, can dramatically alter the overall congestion patterns and quality of the convergence.

9.4.2.2 Quantitative Analysis

This stage of the calibration process is based on direct comparisons between model results and empirical observations, once the DTA is exhibiting a stable and relatively well equilibrated solution. Typically at this stage, the results are relatively stable and

it can be relatively difficult to substantially change the general congestion pattern, i.e., the locations of the queues.

Various statistical measures may be used to quantify the goodness of fit between the DTA output and the observed data, but the actual process of improving the fit by adjusting input data is essentially a manual process based on intuition and modeling judgment, requiring a solid understanding of traffic phenomena and causes of congestion. For instance, understanding how changing a parameter such as link or movement capacity – e.g., by modifying signal timing parameters – affects link travel time, which in turn impacts path choice, is critical to carrying out a calibration exercise, as it makes it possible to predict how certain changes to the inputs should *generally* affect the outputs. Without this predictive insight into the behavior of the model, the process of calibration is little more than trial and error.

Moreover, a model (application) that is not in equilibrium will not necessarily exhibit the expected correlation between changes in travel times and the resulting changes in path choices. From a calibration standpoint, a model that is not in equilibrium is essentially a moving target: since the connection between path travel times and path choices is not reliable, changes to the inputs lead to unexpected and illogical changes in the outputs. This is exactly the problem of artificial instability discussed in Section 9.4.1, *Calibration and Stability*.

In its simplest form, the quantitative analysis stage consists of investigating one or more outliers at a time in order to determine how the model inputs need to change in order to better approximate the observed conditions (empirical data), without degrading the goodness of fit of the other observations. Generally speaking, the sources of error can be broken down into three categories:

– supply side: network and signal timing parameters
– routing: assignment model is not capturing driver behavior
– demand: inaccurate values in the O–D matrix

The remaining sub-sections provide an overview of the general process of investigating outliers and drawing conclusions about the most likely sources of error through some simple examples.

9.4.3 Traffic Flow Calibration

Dynameq automatically collects a wide variety of measures, which can be visualized in various ways, in order to interpret simulation results for network links, turning movements, intersections (nodes), and individual lanes. Evaluation of model outputs typically starts with animating temporal link-based results on the network plot, which provides an overview of the overall traffic conditions and allows the key bottlenecks to be quickly identified. A snapshot of such a plot is shown in Fig. 9.4. This plot displays link flows as bar widths and level of congestion – represented by

Fig. 9.4 Link outflow (width) and occupancy (color)

a measure called *occupancy* – by color, as indicated by the legend. Occupancy is a unitless measure that is a normalized value of link density (which is itself expressed in vehicles/km), and ranges from zero (no vehicles) to 100% (link entirely full of vehicles standing still). Starting from this high-level view, key locations such as bottlenecks can be identified and then investigated further by examining a variety of detailed measures.

The area inside the red oval in Fig. 9.4 indicates heavy congestion in the eastbound direction approaching the north-south freeway. Specifically, there are traffic counts for the link colored red (indicating very high occupancy, in the middle of the oval). This is the eastbound approach to a four-legged intersection, and thus has three exiting movements. The traffic counts at this intersection indicate that the traffic flow on the through movement is considerably lower in the model than observed in the field, while the left and right-turn movements correspond very well with the field data. A typical investigation might proceed as follows.

Figure 9.5 shows the time series plots of link outflow and occupancy measures, which in this case indicate a relatively constant outflow accompanied by increasing

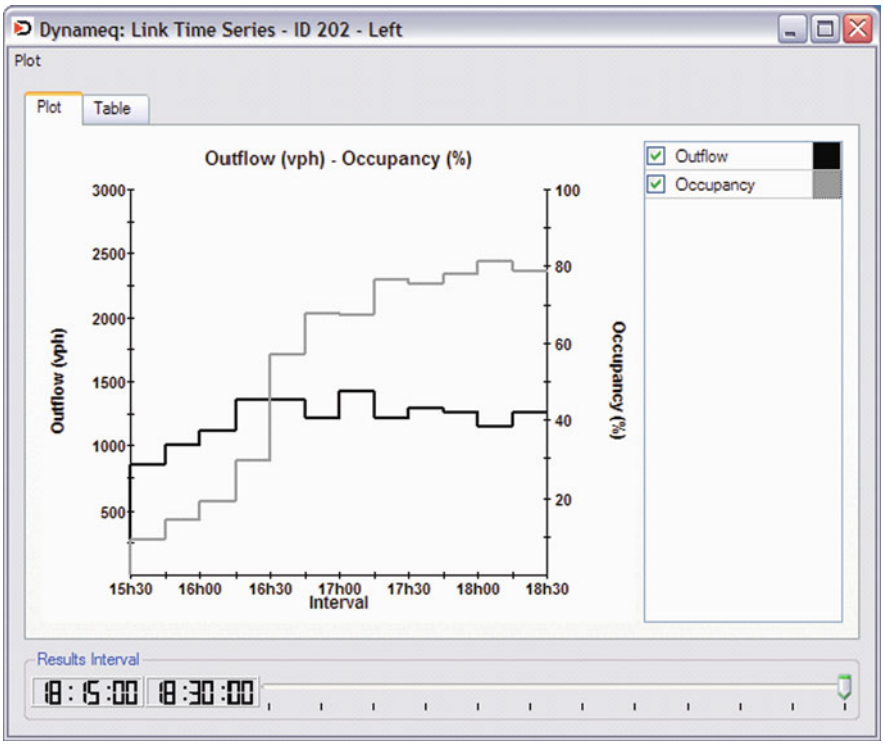


Fig. 9.5 Link outflow and occupancy at a bottleneck

congestion (occupancy). The occupancy plot exhibits a sudden increase at around 16:30. The link occupancy is further analyzed by breaking it down to see the relative contributions to this value due to the vehicles destined for each of the three exiting movements from this link (referred to here as *movement occupancy*), as shown in Fig. 9.6. This plot is characterized by a rapid increase in the number of vehicles destined for the left-turn movement, with a jump at 16:30, followed by a significant increase in the number of vehicles destined for the through movement.

As these occupancy values are known to be reflective of congested conditions, it is quite likely that the increase in queued vehicles for the left turn may in fact be responsible for the increase in queued vehicles for the through movement: this typically occurs when a left-turn pocket overflows and begins to block a regular lane that services the through movement. It is interesting to note at this point the individual outflow values by turning movement, shown in Fig. 9.7: although the left turn movement is responsible for the majority of queued vehicles on the link, the outflow (expressed as a flow rate, in vehicles per hour) of this movement is a fraction of that of the through movement, indicating a major discrepancy in the demand/supply relationships of these two turning movements.

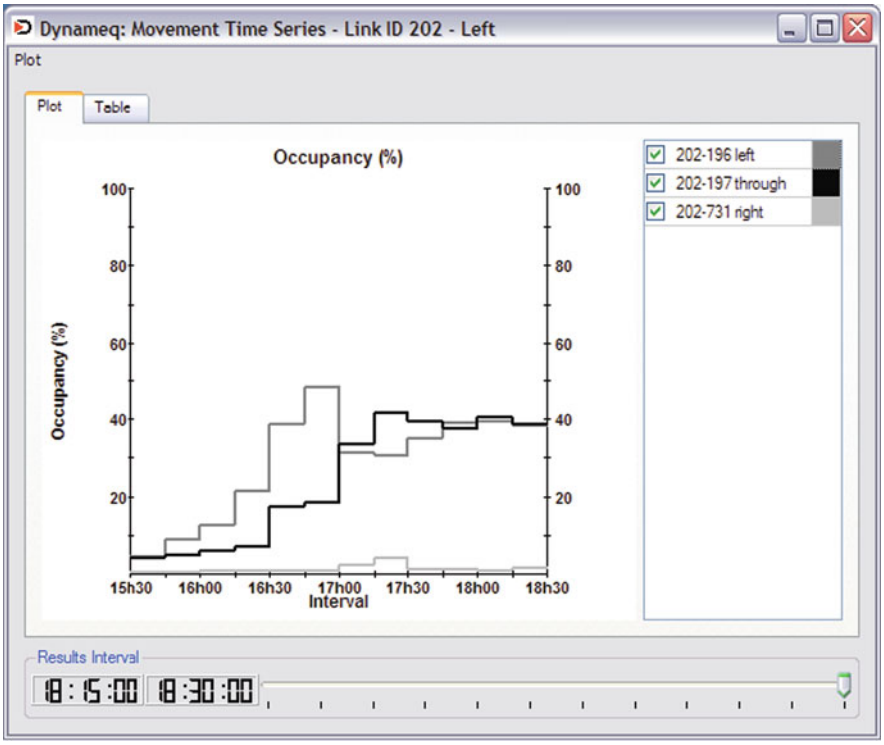


Fig. 9.6 Link occupancy by turning movement at a bottleneck

The situation is further investigated by observing flow rates exiting the link per lane, as shown in Fig. 9.8. The lanes are numbered from the outside edge to the inside, so that lane 3 represents the left-turn pocket. It can be seen that the flow on lane 3 is relatively constant but very low. For the other two lanes, which service the through and right-turn movements, the flows are essentially equal up until 16:30, at which time they diverge rapidly, with the flow on the middle lane dropping while the flow on the outside lane (lane 1) increases. Since this change in

lane-based flow occurs simultaneously with the sharp increase in the number of vehicles queued (occupancy) for the left-turn movement (Fig. 9.6), the cause of the insufficient traffic volume on the through movement is now easy to see.

At 16:30, the left-turn pocket overflows and is blocking lane 2 (the middle lane). As a result, the outflow on lane 2 drops, and the through traffic entering this link begins changing lanes to get around this blockage; as a consequence the flow on lane 1 suddenly increases. Nevertheless, as the through-movement traffic is significantly lower than expected, it must be concluded that the expected flow rate cannot be attained if the left-turn pocket is regularly spilling back and blocking lane 2. Thus, either the demand for this left turn is too high, or the supply, primarily determined by the signal phase design and timing parameters, is too low.

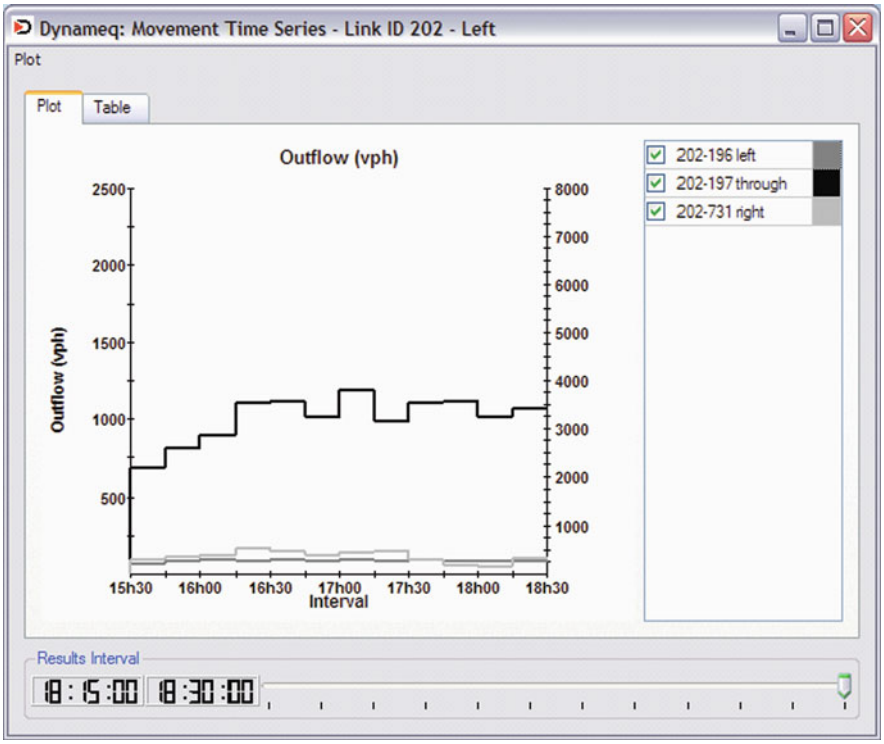


Fig. 9.7 Link outflow by turning movement at a bottleneck

As mentioned above, the field count for the left-turn movement is in agreement with the model. Under congested conditions, the count reflects the supply (capacity) of the movement, and not the demand, and thus says nothing about the correctness of the demand for this movement or the associated queueing. However, the empirical count validates the left-turn capacity in the model, thereby allowing a clear conclusion to be drawn: the demand for the left-turn movement must be too high.

This kind of analysis provides insights into the detailed workings of the traffic flow in the model, and often allows precise conclusions to be drawn about the causes of discrepancies between model outputs and field observations. In many cases, the conclusion may be that the supply is either insufficient or excessive, rather than that the demand is incorrect as in the example above. For such cases, Dynameq has link-based parameters and gap acceptance parameters that can be adjusted locally for the purpose of calibration.

Two link-based parameters are available which are specifically intended for calibration, called the *response time* factor and the *effective length* factor. These are scalar multipliers that are applied to each vehicle during its journey along the link, and can be used to obtain desired values of maximum flow (or *capacity*, in vehicles

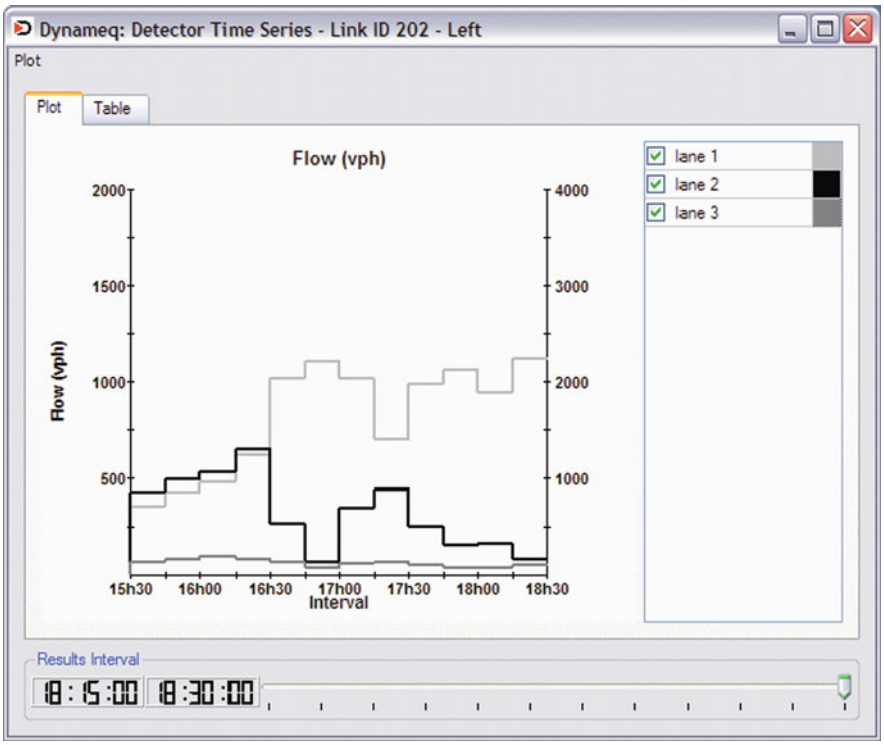


Fig. 9.8 Link outflow by lane at a bottleneck

per hour) and maximum density (in vehicles per kilometer or mile). The free-flow speed of the traffic on the link is another user-defined parameter. Together, these three parameters allow the user to define the speed–flow–density relationship (or *fundamental diagram*) for each link in the network (this relationship was briefly mentioned in Section 9.2, *Core Traffic Flow Models*). As mentioned above, effective capacities under conditions of heavy lane changing or weaving can be difficult to capture without some adjustment to the default inputs: in this situation, the response time factor can be adjusted to account for the fact that drivers often carry out lane-changing maneuvers with lower headways than typically used when traveling behind each other in a single lane.

As mentioned in Section 9.2, *Core Traffic Flow Models*, Dynameq employs a two-parameter gap acceptance model. These two parameters (*critical gap* and *critical wait*) may be adjusted, if desired, at the level of each pair of conflicting movements at an intersection. This makes it possible to account for local effects, such as grade and visibility, on gap acceptance behavior, e.g., how drivers merge at a freeway on-ramp. Default values for various standard situations, such as stop and yield signs, roundabouts, and signalized intersections, are provided by the software.

9.4.3.1 Advanced Modeling Features

A feature that can be helpful in calibration, though is more generally used as a modeling tool for representing special situations, is referred to as *time-varying attributes*. This allows network properties (supply-side data) to change at pre-defined points in time. This can be used to represent various real-world situations, such as congestion pricing (tolls) that changes several times during a peak period, as is currently implemented in Stockholm (Sweden), and London (UK). This can also be used to approximate traffic control or management measures that are triggered by congestion levels reaching a certain threshold, since these conditions typically occur at about the same time everyday. Examples of such measures include variable speed limits on freeways, and opening hard shoulders for regular traffic in order to increase capacity upstream of a major off-ramp. This feature is also useful when there is congestion spilling back into the area being modeled from outside the network: by measuring the typical flow rates in the field during the study period, the flow capacity of an exiting link (called a connector) can be set to follow a time-varying pattern of effective capacity, rather than simply representing the theoretical capacity under ideal conditions.

9.4.4 Route Choice Calibration

Dynameq offers several features specifically for the purposes of evaluating and calibrating route choice behavior, including path display, select link analysis, and generalized cost assignment. Although empirical data about route choice is generally not available for typical applications, inspection of routes can often provide valuable information for the calibration process, and can help to further clarify whether discrepancies between model outputs and empirical data are primarily due to demand-side or supply-side errors. The tool primarily used in addressing this question is *select link analysis*.

Recalling the example of the over-saturated left turn discussed above, and the conclusion that the cause was due to excessive demand rather than insufficient supply, the next step in the analysis is to determine whether the excess of cars for this movement is due to erroneous routing or excessive demand specified in the O–D matrix. The first step in addressing this question is to execute a select link analysis on the turning movement (i.e., a *select-turn* analysis). This procedure identifies all paths which use this turn and provides various outputs associated with these paths, including the corresponding partial O–D matrix. These outputs also include select link simulation results, which are the same types of outputs used in the above example (Fig. 9.4), but counting only those vehicles on the paths that go through the turn in question.

Figure 9.9 shows a snapshot at a given time interval (roughly the middle of the simulation, in order to be fairly representative) of the *select-turn* link flows as bar widths, i.e., representing only those vehicles that use this particular left turn somewhere along their journeys. The plot shows that all of the traffic is destined

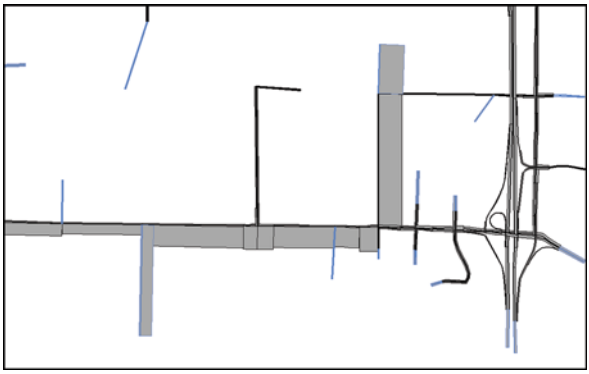


Fig. 9.9 Link flows: select-turn analysis on left turn

to a single destination (situated to the north of the main arterial), about half of the demand comes from a single origin (situated to the south of the arterial just upstream) and the remainder comes from several origins further upstream along this arterial. If necessary, this information can be complemented with path displays showing the alternative paths adopted by other vehicles for the same O–D pairs as those identified by the select link analysis. The conclusion in this case is that the paths using the over-saturated left turn are reasonable and realistic, i.e., there are no preferable alternate paths that these vehicles should be using instead. This observation then leads directly to the conclusion that the excessive demand for this turning movement is attributable to the O–D matrix rather than to the assignment (route choice).

In a situation where the discrepancy is due primarily to the route choice itself, *generalized cost* can be used to "calibrate" the route choice model by considering factors in addition to travel time. Although no detailed examples are provided here of such an application, one recent case involved a calibration exercise for the network of Lausanne, Switzerland, which included a very old part of the city with narrow cobblestone streets. The initial travel time based DTA was clearly routing too much traffic through this area, and this was handled by adding *perceived costs* to these links to make them less attractive. The costs were adjusted manually until an acceptable fit was obtained in this area between the model outputs and empirical data, which consisted primarily of link-based traffic counts.

9.4.4.1 Advanced Modeling Features

Generalized cost assignment is a feature that has uses well beyond that of calibrating a model against empirical data. In some cases, cost formulas and weights for time, distance and direct monetary cost are established as a modeling standard, rather than using a pure travel time-based assignment. The use of tolls in an application clearly requires the use of a generalized cost assignment, and the implementation of time-varying tolls necessitates the use of a dynamic model. Time-varying tolls, and other new tolling mechanisms such as HOT (high-occupancy/toll) lanes with

congestion-dependent pricing, are good examples of the general trend toward the use of increasingly complex traffic management mechanisms, which require the higher level of detail and realism offered by simulation-based DTA.

9.4.5 Calibration – Future Directions

The development of algorithms for calibrating DTA and traffic simulation models has been an important area of research for some time already, though practical and robust (i.e., deployable for use by practitioners) methods for large-scale congested networks are not yet available. The simple example illustrated above demonstrates the underlying complexity of the problem: the data indicated an issue (too little flow) for a particular turning movement, but after analyzing the situation, the actual cause was in fact related to another turning movement for which the empirical data corresponded perfectly well with the model outputs. Another situation that commonly arises, associated with a supply-side (network coding) rather than demand-side error, is a false congestion point that artificially increases travel time on one route, and thereby increases the traffic flow (via the equilibrium-seeking iterative algorithm) on alternative routes. The cause of this problem can be particularly challenging to identify if the empirical data does not cover the false bottleneck, but only the alternative routes. Despite these challenges, which apply both to manual calibration and the ongoing improvement of automated calibration tools, real-world applications of DTA are being successfully calibrated to reasonable thresholds of goodness-of-fit, and such applications are becoming increasingly common.

9.5 Selected Applications

This section presents an overview of a typical DTA application that was recently carried out using Dynameq. The modeled area is the entire Municipality of Ljubljana, Slovenia. The description below briefly presents the objectives of the modeling study and the results of the base year calibration work. Applying the calibrated model to evaluating various improvements and alternatives is currently underway. The section ends with a summary of software performance metrics for this and a few other recent projects.

The city of Ljubljana, including the surrounding suburbs, represents the highest level of urbanisation in Slovenia and is the most important central town. Ljubljana lies in the heart of the Central Slovenian Region which has roughly half a million inhabitants, of which 54% (268,000) live in the Municipality of Ljubljana. Of the 215,000 jobs in this region, 77% are in Ljubljana.

Ljubljana is also the main national traffic node, at the junction of the of the primary road and railway flows in the country. The road network has a traditional radial

Fig. 9.10 Ljubljana network with satellite photo

pattern encircled with a ring road, the area inside which is approximately 60 square kilometers (see Fig. 9.10). As with most densely populated urban areas, congestion is continuously increasing due to growing traffic demand and is expected to worsen despite the fact that the population is not expected to grow significantly in the future. Moreover, the current arrangement of public transport fails to provide a viable alternative to the private car for most trips, and the share of public transport trips is falling. The most popular, and cost effective, means of mitigating congestion and increasing mobility are focused around traffic management strategies and increasing public transportation, rather than building new road capacity into the network.

Future traffic related projects are based on two main strategies:

– implementing transit priority on the main radial arteries;
– increasing traffic capacity on the ring road.

Increasing vehicular capacity on the radial arteries is purposely being avoided. Because the radial pattern has the effect of focusing traffic through the city center, the congestion level in this area is very sensitive to the capacities of these arterial routes. Increasing the capacities of the latter could easily result in overloading the city center with vehicles, leading to even more congestion and less mobility.

Since many traffic management strategies cannot be modeled accurately using traditional static assignment (travel forecasting) models, it was decided to adopt a DTA approach for evaluating these policies. The main policies being considered include:

– reserved lanes for buses on city arterials;
– transit priority at intersections;
– park and ride locations;
– increasing capacity on the ring road (expanding to 6 lanes);
– reconstruction of many city roads;
– new ITS measures for improving capacity, safety, driving comfort and environmental impacts (the ring road is already under ITS management);
– road pricing for the central (downtown) area.

The modeled area is the entire Municipality of Ljubljana, covering approximately 274 km^2 and comprising the entire national and municipal road networks. This includes approximately 550 km of roadway (bi-directional): the model itself has 1280 km of directional roadway representing over 1500 lane-kilometers. Figure 9.10 shows the modeled network drawn in white overlaying a satellite image of the area. The model includes private vehicles, public and freight transport. Public passenger transport includes interurban, suburban and urban bus lines. The model was calibrated to the morning and afternoon peak periods using 2008 data.

Due to the high level of detail required for capturing traffic phenomena such as the effects of bus delays on overall congestion, and the effect of transit signal priority on bus travel times, it was decided to adopt a high-fidelity DTA model using a microsimulation approach for traffic flow modeling. Figure 9.11 shows a snapshot of the link flows and occupancies in the network at 4:30 p.m. Traffic is flowing relatively well on the ring road, as indicated by the wide bars and blue colors: dark blue (occupancy < 15%) indicates free-flowing conditions, while light blue (15% < occupancy < 30%) indicates locations that are essentially at capacity, but not yet congested. A number of critical bottlenecks, indicated by yellow, orange and red, can be seen along the radial arteries.

The calibration effort was supported by an extensive survey of traffic counts, comprising 564 link and turning movement count locations, including 84 on the highways (including the ring road) and 154 on the main city streets. Figures 9.12 and 9.13 show scatter-plots of the model results (y-axis) vs. traffic counts (x-axis) for the AM and PM peak periods, respectively, for the entire set of count locations. Figures 9.14 and 9.15 show the corresponding AM and PM plots, respectively, for the highway (including ring road) locations, while Figs. 9.16 and 9.17 show the AM and PM plots, respectively, for the main city roads. Regression analysis was used to produce a linear best-fit for each data set, as shown on the plots. Table 9.1 presents the R^2 statistics for the linear regression analysis, with values ranging from 0.94 to 0.96. These are considered to be very good results, particularly for a network of this size. Although the numeric values of the slopes are not reported, they can be seen from the plots to be generally just below or around 1, as expected.

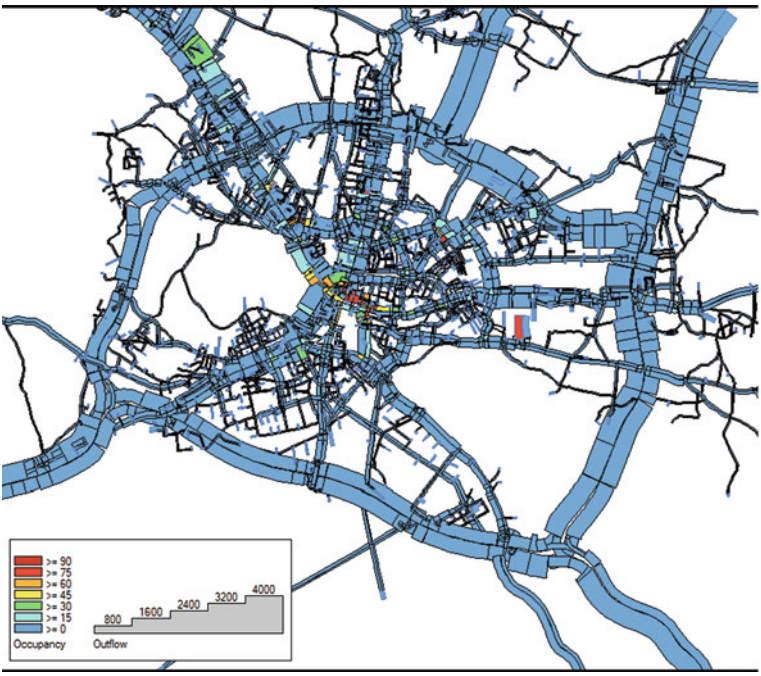


Fig. 9.11 Link outflow and occupancy in Ljubljana model at 16:30

Tables 9.2 and 9.3 show travel time comparisons for some of the main routes in the network for the AM and PM scenarios, respectively. These routes normally consist of a sequence of several network links and the empirical data is collected by actually driving the routes several times during the peak period and taking the average measured travel time. The reported travel times are rounded to the nearest minute, while the percentage differences are computed based on the exact values. The goodness of fit of the travel time results were found to be excellent: the relative differences for the AM paths were between 5% and 8%, while the PM results had similar values, but with one path at 12%.

It should also be mentioned that this calibration was carried out without the use of matrix adjustment algorithms or techniques. Matrix adjustment algorithms, which automatically adjust the demand matrix in order to provide a better fit to a set of traffic counts, have been available for many years for static assignment models and can be used to pre-process the demand matrices for a DTA model as well. Their use poses some difficulties in the context of long-term planning studies for which future demand scenarios must be modeled, since future traffic counts are not available for adjusting those matrices. Avoiding matrix adjustment in these cases maintains a stronger linkage between the DTA results and the synthetic demand model.

Some basic software performance metrics for this project, as well as a few other recent projects, are presented in Table 9.4. These include the following:

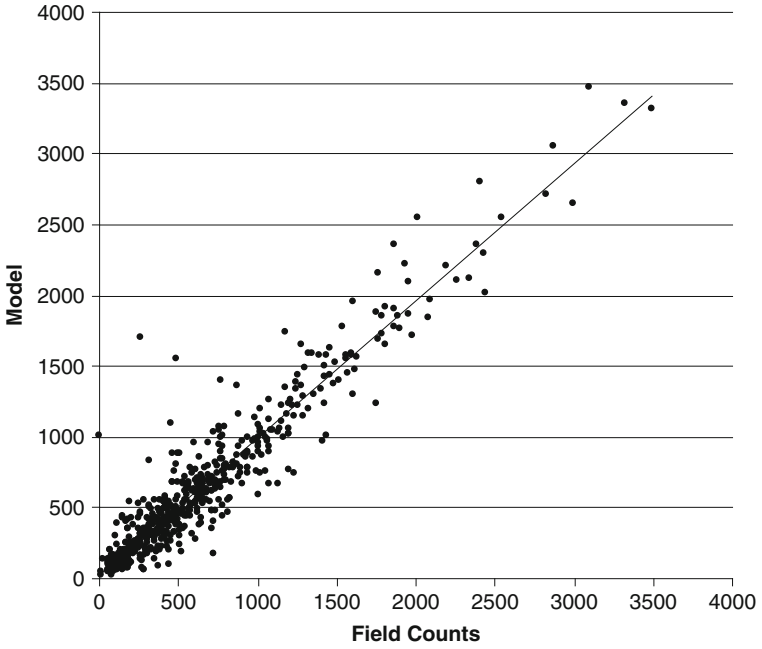


Fig. 9.12 Model vs. field data for all roads: morning peak hour

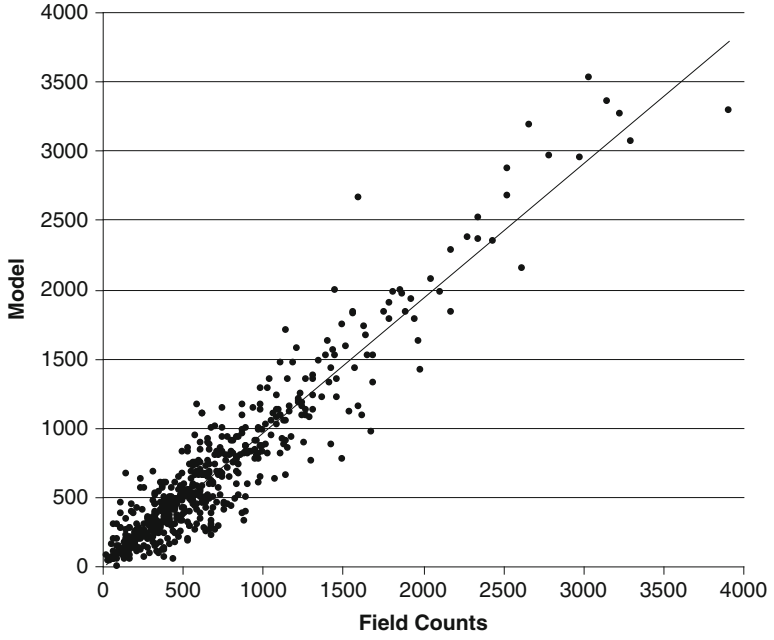


Fig. 9.13 Model vs. field data for all roads: afternoon peak hour

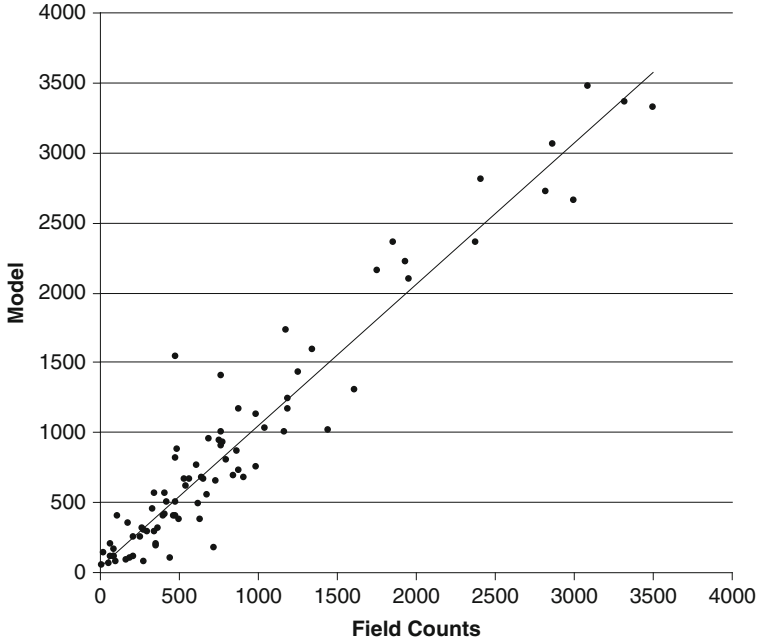


Fig. 9.14 Model vs. field data for highways: morning peak hour

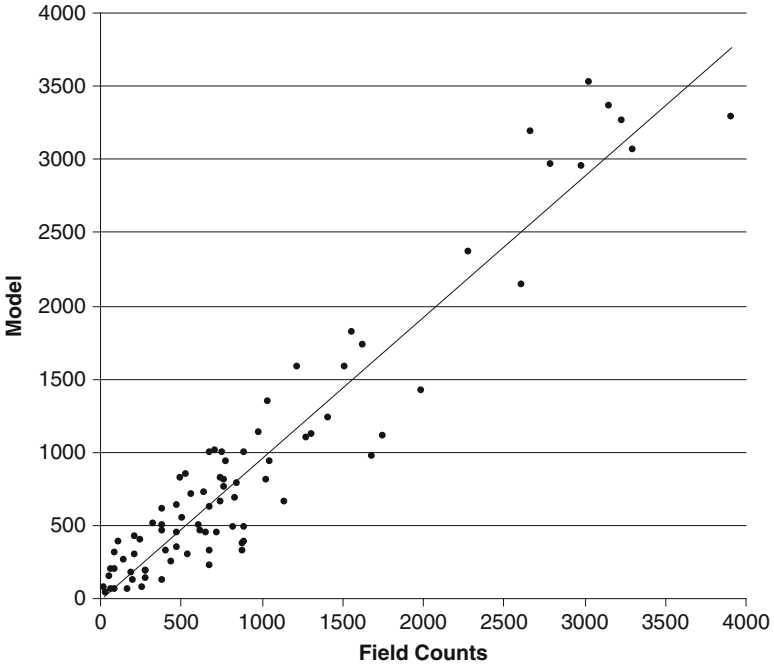


Fig. 9.15 Model vs. field data for highways: afternoon peak hour

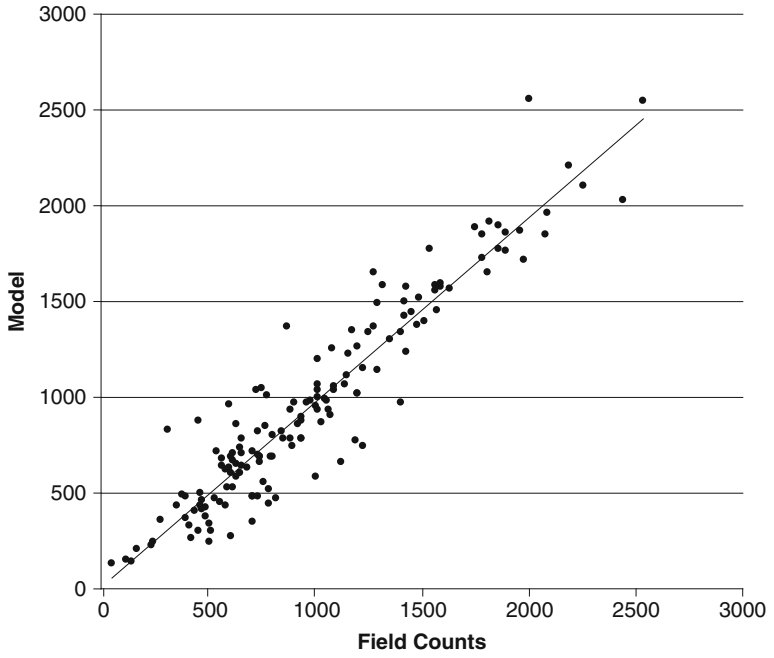


Fig. 9.16 Model vs. field data for main city roads: morning peak hour

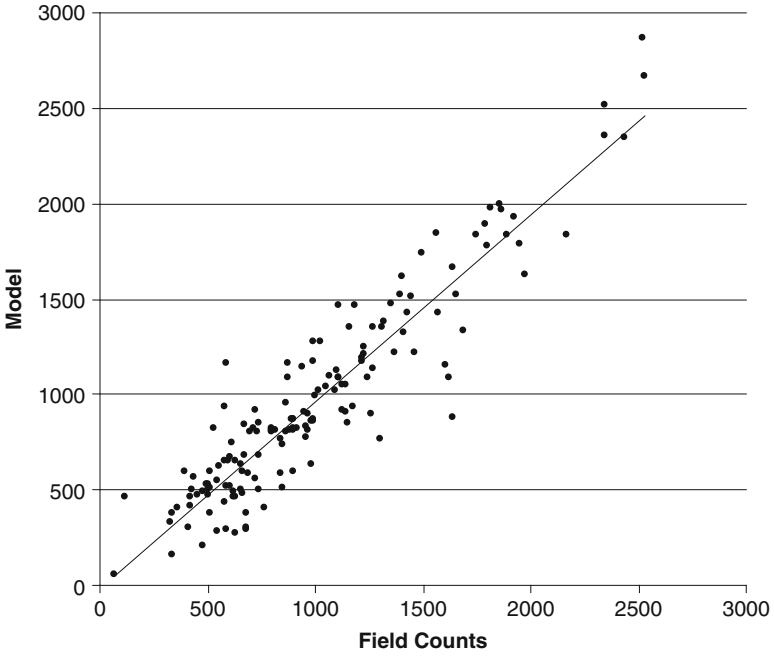


Fig. 9.17 Model vs. field data for main city roads: afternoon peak hour

Table 9.1 R^2 statistics for linear regression against traffic counts

All vehicles	Number of field counts	Morning	Afternoon
All roads	564	0.95	0.95
Highways	84	0.96	0.96
Main city roads	154	0.95	0.94

Table 9.2 Travel time comparisons for AM peak hour

	Average travel time		
Path	Survey (min)	Model (min)	Relative difference (%)
Path 1 (Vič, Smelt)	15	16	7
Path 2 (Rožna dolina, Vojkova c.)	19	19	5
Path 3 (Črnuče, Vokova c.)	13	12	7
Path 4 (Brezovica, Vojkova c.)	18	17	5
Path 5 (Zaloška c., Vojkova c.)	13	13	8
Path 6 (priključek LJ-jug, Vojkova c.)	17	17	6

Table 9.3 Travel time comparisons for PM peak hour

	Average travel time		
Path	Survey (min)	Model (min)	Relative difference (%)
Path 1 (Vojkova c., Jadranska)	14	13	7
Path 2 (Vojkova c., BTC – City Park)	14	13	7
Path 3 (Vojkova c., Rožna dolina)	14	14	7
Path 4 (Vojkova c., Črnuče)	12	12	8
path 5 (Vojkova c., Zaloška c.)	10	12	12
Path 6 (Vojkova c., priključek LJ-jug)	19	17	5

– size of the network measured in number of links;
– duration of the modeled study period;
– total volume of vehicles summed over all classes;
– number of iterations used in calibrated base year DTA;
– average relative gap (final iteration) in calibrated base year DTA;
– average CPU time (minutes) per iteration in calibrated base year DTA;
– real-time speed up: study period duration divided by the portion of CPU time associated with traffic flow simulation (last iteration).

Metrics reported for projects that are currently underway or that were run only as test networks, and thus are not fully calibrated, do not include the number of iterations or relative gap. CPU times were obtained on a DELL Optiplex 755 running

Table 9.4 Software performance metrics for various networks

Location	Ljubljana, Slovenia	Tel Aviv, Israel	Montreal, Canada	Manchester, UK (motorway)	San Francisco, USA
Number of links	8466	3603	8062	2009	33,487
Study period (h)	3	3	3	1	3
Total volume (vehicles)	174,247	199,049	399,306	146,898	550,000
Iterations	125	120	75	na	na
Average relative gap (%)	3.0	6.0	3.7	na	na
CPU/iteration (min)	2.5	2.1	4.8	1.6	12.5
Simulation speed-up	83×	92×	43×	40×	20×

the Windows Vista™ Business (32-bit) operating system, with a 3.0 GHz processor and 3.325 Gb RAM. The exception is the San Francisco network, which was run under a Linux operating system on a 2.6 GHz processor.

DTA models are a relatively new tool to arrive into traffic planning and engineering practice. The general experience with DTA has been that due to the combination of the scale of these models and the relatively high sensitivities they exhibit, i.e., compared to static assignment models traditionally used for travel forecasting, meeting conventionally accepted goodness-of-fit calibration criteria can sometimes be challenging. The overall quality of the calibration results presented in this section are considered to be excellent and are very promising for the continued use and adoption of simulation-based DTA for projects of similar size and scope.

Acknowledgment The authors would like to thank PNZ Consulting Designing Ltd., who are carrying out the Ljubljana project, for generously providing the related information presented above.

References

Ben-Akiva M, Koutsopoulos HN, Mishalani R (1998) "DynaMIT: A simulation-based system for traffic prediction". Paper presented at the DACCORD short term forecasting workshop, Delft, The Netherlands. See also http://web.mit.edu/its/products.html, accessed 12 September 2009

Brackstone M, McDonald M (1999) Car-following: a historical review. Trans Res 2F(4):181–196

Chevallier E, Leclerq L (2009) Do microscopic merging models reproduce observed priority sharing ratio in congestion? Trans Res 17C:328–336

Chiu Y-C, Bottom J, Mahut M, Paz A, Balakrishna R, Waller T, Hicks J (2010). DTA Primer, Network Modeling Committee (ADB30) of the Transportation Research Board, Washington, DC. http://www.nextrans.org/ADB30/UPLOAD/ssharma/dta_primer.pdf

Dafermos SC (1971) An extended traffic assignment model with application to two-way traffic. Trans Sci 5, 366–389

Diakaki C, Papageorgiou M (1996) Integrated modeling and control of corridor traffic networks using the METACOR modeling tool. Dynamic systems and simulation laboratory, Technical University of Crete. Internal Report No. 1996-8. Chania, Greece, p 41

Florian M, Mahut M, Tremblay N (2008) Application of a simulation-based dynamic traffic assignment model. Eur J Oper Res 189(3):1381–1392

Friesz T, Bernstein D, Smith T, Tobin R, Wie B (1993) A variational inequality formulation of the dynamic network user equilibrium problem. Oper Res 41:179–191

Gabard JF (1991) Car-following models. In: Papageorgiou M (ed) Concise encyclopedia of traffic and transportation systems, Pergamon Press, Oxford, pp 337–341

Hoogendoorn SP and Bovy PHL (1999) Macroscopic multiple user-class traffic flow modelling: a multilane generalisation using gas-kinetic theory. Proceedings of the 14th international symposium on transportation and traffic theory. Jerusalem, Israel, 20–23 July 1999

Lawphongpanich S, Hearn DW (1984). Simplicial decomposition of the asymmetric traffic assignment problem. Transport Res B 17:123–133

Leonard DR, Gower P, and Taylor NB (1989). CONTRAM: Structure of the model. Transport and Road Research Laboratory (TRRL) Research Report 178. Department of Transport, Crowthorne.

Leventhal T, Nemhauser G, Trooter L Jr (1973). A column generation algorithm for optimal traffic assignment. Transport Sci 7:168–176

Lighthill MJ, Whitham GB (1955) On kinematic waves I: flood movement in long rivers. II: a theory of traffic flow on long crowded roads. Proc R Soc Lond, A229:281–345

Luenberger D (1984) Linear and nonlinear programming, 2nd edn. Addison-Wesley, Inc., Reading, Massachusetts

Mahmassani HS, Abdelghany AF, Huynh N, Zhou X, Chiu Y-C, Abdelghany KF (2001). DYNASMART-P (version 0.926) User's Guide. Technical Report STO67-85-PIII, Center for Transportation Research, University of Texas at Austin, Austin, USA

Mahut M (1999) Speed maximizing car-following models based on safe-stopping rules. Compendium of Papers CD-ROM, 78th Annual Meeting of the Transportation Research Board, January 10–14, 1999, Washington, DC

Mahut M (2001) Discrete flow model for dynamic network loading. Ph.D. thesis, Département d'informatique et de recherhe opérationelle, Université de Montréal

Mahut M, Florian M, Tremblay N (2008) Comparison of assignment methods for simulation-based dynamic-equilibrium traffic assignment. Compendium of Papers CD-ROM, 87th Annual Meeting of the Transportation Research Board, January, 2008, Washington, DC

Messmer A (2000a) METANET a simulation program for motorway networks (documentation). Dynamic Systems and Simulation Laboratory, Technical University of Crete, Chania, Greece

Messmer A (2000b). METANET-DTA an exact dynamic traffic assignment tool based on METANET. Dynamic Systems and Simulation Laboratory, Technical University of Crete, Chania, Greece, p 37

Nagel K, Schreckenberg M (1992) A cellular automaton model for freeway traffic. Journal de Physique I France, 2:2221–2229

Newell GF (1993) A simplified theory of kinematic waves in highway traffic. Part I: general theory. Transport Res B 27B(4):281–287

Newell GF (2002) A simplified car-following theory: a lower order model. Transport Res Part B 36B(3):195–205

Papageorgiou M (1990) Dynamic modelling, assignment and route guidance in traffic networks. Transport Res 24B(6):471–495

Patriksson M (1994) The traffic assignment problem: models and methods. Topics in Transportation, VSP BV, Utrecht, The Netherlands

Richards PI (1956) Shock waves on the highway. Oper Res 4:42–51

Rosen JB (1960) The gradient projected method for nonlinear programming, part I: linear constraints. J Soc Indus Appl Math 8:181–217

Van Aerde M (1999) INTEGRATION Release 2.20 for Windows: User's Guide. MVA and Associates, Kingston, Canada

Wardrop JG (1952) Some theoretical aspects of road traffic research. In: Proceedings of Institue of Civil Engineerings, Part II, vol 1. pp 325–378

Ziliaskopulos AK, Lee S (1997) A cell transmission based assignment-simulation model for integrated freeway/surface street systems. Transport Res Rec 1701:12–22

Chapter 10
Traffic Simulation with DynaMIT

Moshe Ben-Akiva, Haris N. Koutsopoulos, Constantinos Antoniou, and Ramachandran Balakrishna

10.1 Introduction

DynaMIT (Dynamic Network Assignment for the Management of Information to Travelers) is a simulation-based Dynamic Traffic Assignment (DTA) model system that estimates and predicts traffic conditions. Its development was funded by the US Department of Transportation's Federal Highway Administration for traffic information generation for Advanced Traveler Information Systems (ATIS). However, providing current (instantaneous) information to travelers may worsen network performance as it transfers congestion from one location to another, especially when market penetration of ATIS is high. DynaMIT provides predictive information, consistent with the conditions drivers will experience in the network.

DynaMIT includes detailed models of travel demand and behavior, network supply, and their complex interactions. The system is designed to interface with a surveillance system in real-time, estimate current network conditions, and generate short-term predictions for future conditions. These predictions support the operation of ATIS, to aid drivers in making informed route and departure time choices, and serve traffic management centers in functions such as emergency response, and traffic management.

M. Ben-Akiva (✉)
Massachusetts Institute of Technology, 77 Massachusetts Ave., Rm. 1-181, Cambridge, MA 02139, USA
e-mail: mba@mit.edu

H.N. Koutsopoulos
The Royal Institute of Technology, KTH Teknikringen 72, SE – 100 44 Stockholm, Sweden
e-mail: hnk@infra.kth.se

C. Antoniou
National Technical University of Athens, 9 Heroon Politechniou st., Athens GR-15780, Greece
e-mail: antoniou@central.ntua.gr

R. Balakrishna
Caliper Corporation, 1172 Beacon Street, Suite 300, Newton, MA 02461, USA
e-mail: rama@caliper.com

J. Barceló (ed.), *Fundamentals of Traffic Simulation*,
International Series in Operations Research & Management Science 145,
DOI 10.1007/978-1-4419-6142-6_10,

DynaMIT's demand and supply model components provide a modular system that easily adapts to different objectives. DynaMIT's primary application is route guidance in real-time settings. An off-line version for short-term planning applications was also developed (Ben-Akiva et al., 2001). Both applications require the modeling of demand, supply, and their interactions. The details of the interactions may be altered for specific applications. The real-time DynaMIT (DynaMIT-R) interfaces with the surveillance system, obtains sensor data as they become available, estimates and predicts the network state, and generates consistent, anticipatory route guidance. The planning version (DynaMIT-P) is useful for short-term planning applications and establishes dynamic equilibrium conditions. The rest of this chapter focuses on DynaMIT-R. However, the same modeling principles and components are used in DynaMIT-P.

10.2 Model Building Principles in DynaMIT

DynaMIT aims to generate route guidance information, such as travel times disseminated through variable message signs (VMS) and in-vehicle devices, to support the operation of ATIS. In ATIS, anticipatory (prediction-based) route guidance is likely to be more effective than route guidance based on historical or current traffic conditions because it accounts for the evolution of traffic conditions over time and throughout the network. The predictive information is based on forecasts of future traffic situations that will occur at network locations at the time the driver will actually arrive there. DynaMIT generates accurate network state predictions that will drive the generation of route guidance. Figure 10.1 outlines the general DTA framework on which DynaMIT is built on. The guiding principle is that generation of

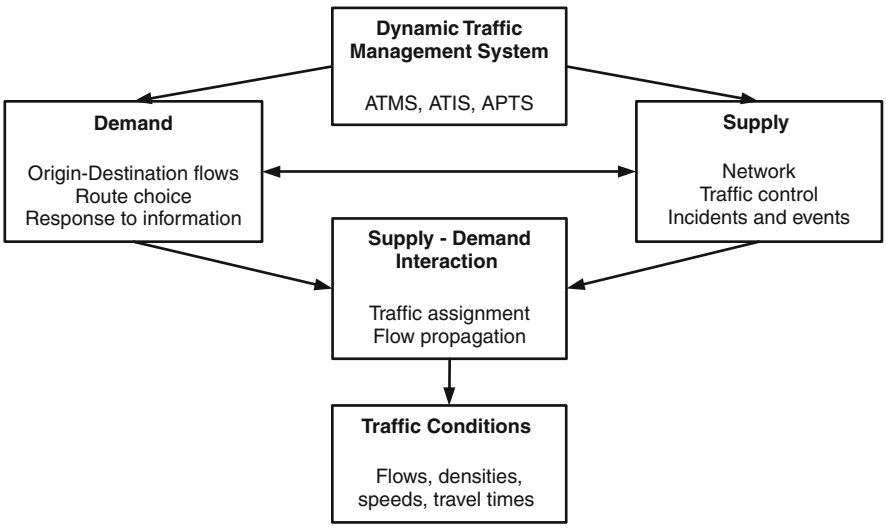


Fig. 10.1 General DTA structure

anticipatory guidance must address a fundamental issue: anticipatory guidance is derived from predictions of future conditions; but these conditions are affected by drivers' reactions to the guidance. It is therefore necessary to ensure the consistency of anticipatory guidance so that the forecasts that lead to the guidance are similar to, within the limits of modeling accuracy, the outcome resulting from drivers reacting to the guidance.

More specifically, DynaMIT is a synthesis of multiple models and algorithms (shown in Fig. 10.2) and uses detailed travel demand and network supply simulators to fuse multiple sources of information and perform state estimation and prediction.

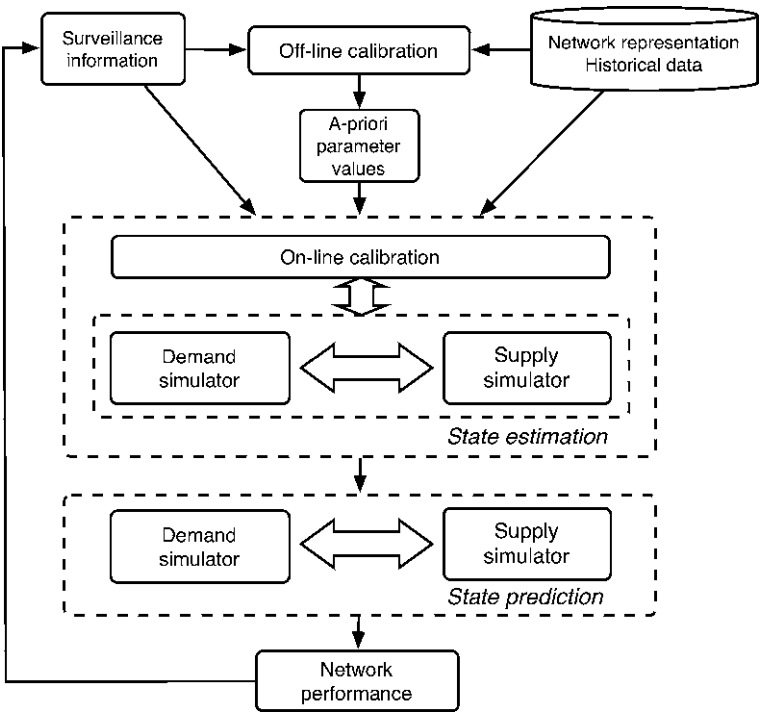


Fig. 10.2 DynaMIT real-time framework

The demand simulator mimics network-wide demand patterns through time-dependent origin–destination (OD) matrices and captures the travel choices of individual motorists (e.g., route choice). Individual travel demand decisions include origin, destination, departure time, and route. Most of these decisions occur before the trip begins. DynaMIT adopts time-dependent origin–destination flows to capture network-wide travel patterns. DynaMIT must also anticipate the response of travelers to the information planned for dissemination (Ben-Akiva et al., 1997). A disaggregate representation of demand that accounts for the individual's socioeconomic characteristics and access to information is used.

The supply simulator is mesoscopic, used to capture traffic dynamics, and evaluate the performance of the network, including formation and dissipation of queues, spillback effects, impacts of incidents, and bottlenecks. It represents traffic dynamics using speed–density relationships and queuing theory (Ben-Akiva et al., 2002).

The complex demand–supply interactions are represented by algorithms that estimate the current network state, predict future conditions, and help in generating anticipatory route guidance and control strategies. An online calibration component allows for the dynamic adjustment of key model inputs and parameters so that the model is consistent with prevailing traffic conditions.

DynaMIT, in real-time applications, operates in a rolling horizon mode, illustrated with an example in Fig. 10.3. At 8 a.m., DynaMIT starts an execution cycle and performs a state estimation using data collected during the last 5 min. When the state of the network at 8:00 a.m. is available, DynaMIT predicts for a given horizon and computes a guidance strategy that is consistent with the prediction. At 8:07 a.m., DynaMIT finishes computing and broadcasts the guidance strategy to the motorists. This strategy will be in effect until a new strategy is generated. At 8:07 a.m., DynaMIT starts a new cycle. The state estimation is now performed for the last 7 min, using all the data the surveillance system collected during that time. DynaMIT will incorporate the new data into its simulation of current network conditions. The new network estimates are used to generate a new prediction and guidance strategy.

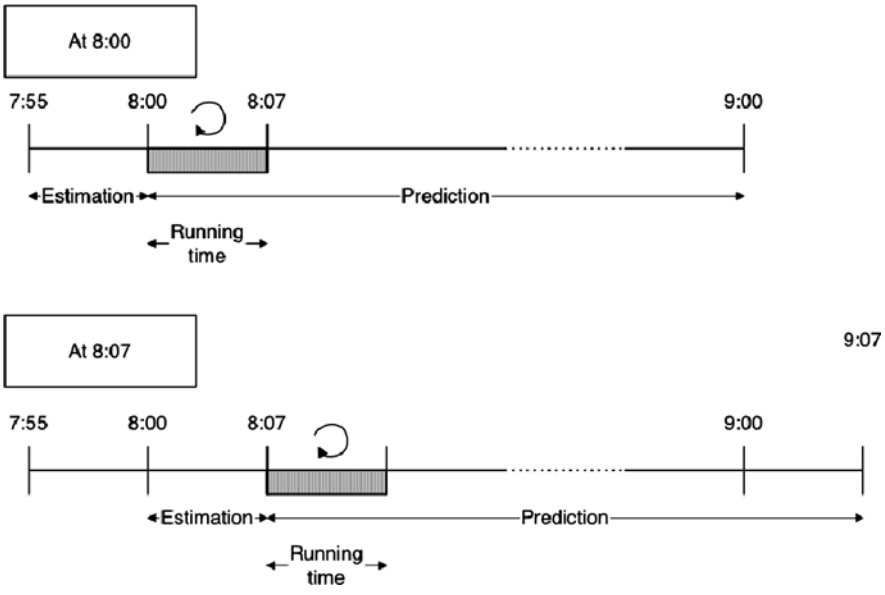


Fig. 10.3 Rolling horizon implementation

10.3 Fundamental Core Models

The main models within DynaMIT are the demand and supply simulators and the online calibration component, which are presented in this section. DynaMIT's graphical user interface (GUI) is also presented.

10.3.1 Demand

DynaMIT employs a disaggregate demand representation to model individual drivers' pre-trip and en route decisions, including response to information. Each individual is considered with his/her socioeconomic characteristics and access to information. An aggregate demand representation in the form of time-dependent origin–destination matrices is also used to estimate and predict network demand levels.

DynaMIT adjusts historical OD flows (which are part of the input database) to capture two phenomena. First, driver responses to real-time information is captured through disaggregate behavioral models that estimate and predict departure time changes, route choice, and route switching (Ben-Akiva and Bierlaire, 2003). Second, the online calibration model is used to adjust the OD matrices based on real-time traffic counts to capture daily demand fluctuations. DynaMIT's demand simulation methodology can be summarized as follows:

$$\mathcal{D} = \mathcal{D}_{\text{hist}} + \Delta\mathcal{D}_{\text{info}} + \Delta\mathcal{D}_{\text{fluct}} + \varepsilon \tag{10.1}$$

where $\mathcal{D}$ is the actual demand; $\mathcal{D}_{\text{hist}}$ is the historical demand; $\Delta\mathcal{D}_{\text{info}}$ represents the influence of traveler information on drivers' behavior; $\Delta\mathcal{D}_{\text{fluct}}$ models systematic yet unobserved modifications in individuals' activity patterns; and ε is a random error term.

The demand simulation process is captured by the five steps shown in Fig. 10.4. The aggregate historical flows are first disaggregated into a habitual list of drivers (step a). These drivers then make various choices that could change destination and route (step b). This updated, disaggregate representation is aggregated back into OD matrices (step c) before the online calibration step (step d) is performed to estimate OD matrices. Finally, the adjusted matrices are disaggregated (step e) to generate the final list of drivers. These steps are discussed below.

(a) Historical information. The first step of the demand simulation is the disaggregation of the historical OD matrices into a historical population of travelers. Travelers are generated off-line and are stored in a database. Each traveler is assigned a vector of socioeconomic characteristics generated by Monte Carlo simulation based on their distributions within the actual population. A habitual travel behavior is assigned to each traveler. Origin–destination and habitual departure times are directly provided by the historical OD matrices. A route choice model based on historical travel times is used to determine the habitual route.

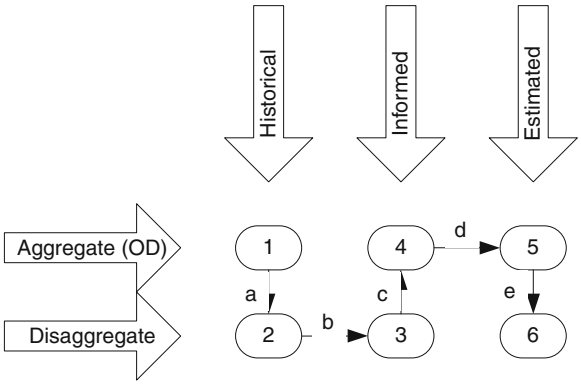


Fig. 10.4 Overview of demand simulation steps (a: disaggregation of historical OD flows, b: travel behavior update, c: aggregation into OD tables, d: online OD estimation and prediction, e: generation of driver population)

(b) Response to information. The validity and relevance of DynaMIT's predictions and generated travel information depends on its ability to capture travelers' decisions, particularly in response to the information they receive. The demand simulator provides the framework to represent this behavior through a number of behavioral models. These models adjust the departure time and the route of each traveler receiving information. Some travelers may cancel their trip or change transportation modes. Travelers' behavior is categorized based on the information type and whether the choice takes place pre-trip or en route. The information types are no information, descriptive information, and prescriptive information. If no information is available, then responses to guidance models are not required since the travelers will not revisit habitual travel patterns. Only habitual models are required. For descriptive and prescriptive information, however, both habitual and response to guidance models are necessary. A path size (PS) logit model (Ben-Akiva and Bierlaire, 1999), which corrects for the IIA property violation common in route choice, is used to provide probabilities for route selection. The probability *P(i)* of a driver choosing route *i* from his/her choice set *C* is given by

$$P(i) = \frac{e^{V_i + \ln PS_i}}{\sum_{j \in C} e^{V_j + \ln PS_j}} \tag{10.2}$$

V_i and V_j are the systematic utilities of routes *i* and *j*, respectively. The systematic utilities in the route choice model are functions of trip attributes including path travel times, path costs, and freeway bias. Depending on the situation, travel times may be habitual or predicted. PS_i and PS_j are the corresponding path *sizes* and capture the correlations between alternatives arising from overlapping path sections. The path size terms capture the effects of overlapping routes on drivers' perceptions of different route alternatives (Ben-Akiva and Bierlaire, 2003).

The pre-trip decisions based on prescriptive information are captured by a compliance model. The pre-trip decisions based on descriptive information are captured by a choice model, where all combinations of departure time interval and route are included in the choice set (Fig. 10.5). The correlation structure of the choice set

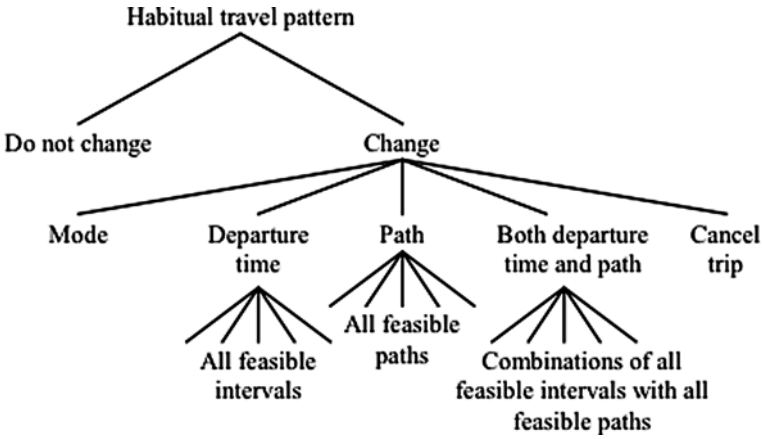


Fig. 10.5 Pre-trip choice set in the case of descriptive information

can be handled by a nested logit or a probit model. The en route models share the same structure as the associated pre-trip models but do not include departure time choices.

(c) Aggregation. Once the behavioral update and response to information is completed, the list of informed drivers is aggregated into the OD table cells, so that the aggregate online demand estimation and prediction step can be performed.

(d) Online OD estimation and prediction. DynaMIT's online calibration model has the capability of doing real-time estimations of the OD flows for the current interval. The details of the estimation and prediction approach are discussed in Section 10.3.3. The model performs short-term predictions of OD flows (for the next 30–60 min). The basic problem of OD prediction is to compute, in real-time, estimates of future OD flows from the current OD estimates and historical OD flows. The OD prediction is based on an autoregressive process, used by the Kalman filtering approach, which provides a prediction tool with real-time capabilities that is consistent with the estimation process. The autoregressive process models the temporal relationship among deviations in OD flows, described in more detail in Section 10.3.3.

(e) Disaggregation – generation of drivers. The estimated and predicted demand matrices need to be disaggregated into a list of drivers before being loaded into the mesoscopic traffic simulator. This process re-uses the off-line generated drivers (step (a)), removing and adding new drivers in each cell as needed (due to the adjustments of the macroscopic process).

10.3.2 Supply

Supply refers to the transportation network and the traffic control system. The network representation consists of static and dynamic components. The static components represent the topology of the network. The dynamic components capture

traffic dynamics. While the static components are fixed during the simulation, the dynamic components can be continuously updated through the online calibration model. The network consists of traditional links, nodes, and loading elements. Each link is divided into segments that capture variations of geometry and traffic conditions along the link. While most segments are defined in advance, additional segments can be dynamically created to capture the presence of incidents. Each segment has a capacity constraint at the downstream end, referred to as the output capacity. Each segment is divided into a moving part and a queuing part. The moving part is the section where vehicles can move with some speed. Queuing parts represent vehicles that are queued up. During queuing, vehicles are assigned to lanes according to their route. A spillback occurs when the downstream segment is blocked (for example, the queue length of the downstream segment is equal to the segment length).

The complexity of the network flows is captured by integrating three classes of models. These classes include capacities associated with roadway features, incidents, and intersection controls; deterministic queuing reflecting the effects of bottlenecks; and macroscopic speed–density relationships reflecting uninterrupted flow.

Capacities. Each segment has a downstream capacity constraint. This capacity constraint can be caused by the physical characteristics of the road, an incident, or a specific control device. The determination of initial values of the output capacities is based on recommendations from the Highway Capacity Manual (HCM, 2000) or may be calibrated using traffic data collected from the field. These values are then dynamically adjusted through the online calibration, described in Section 10.3.3.

Queuing and Moving parts. A deterministic queuing model is used to calculate waiting times at queues. The waiting time calculations are based on the capacity. Each segment has an output capacity and an acceptance capacity. The output capacity defines the rate at which vehicles can leave the segment. The acceptance capacity is the physical space available to store additional vehicles on a segment. When the acceptance capacity is zero, no more vehicles can enter the segment and spillbacks occur.

For the moving part, the following speed–density function is currently used:

$$u = \begin{cases} u_f & \text{if} \quad k < k_{\min} \\ u_f \left[1 - \left(\dfrac{k - k_{\min}}{k_{\text{jam}}} \right)^{\beta} \right]^{\alpha} & \text{otherwise} \end{cases} \tag{10.3}$$

where u denotes the space mean speed, u_f is the free-flow speed, k is the density, $k_{\min}$, k_{jam}, and α and β are model parameters. Initial values of these parameters can be calibrated off-line and then adjusted online.

Simulation process. The simulation of traffic operations is time-based. It proceeds in two phases: the *update phase* and the *advance phase*. During the update

phase, the calculations are the most time consuming. The traffic dynamics parameters (e.g., densities and speeds) used in the simulation are updated. The calculations in the advance phase operate at the microscopic level. The vehicles are advanced to new positions. The advance phase has a higher frequency than the update phase. The time intervals, for the update and advance phases, depend on the application and are selected based on the best compromise between accuracy (e.g., network travel times) and computational performance.

In addition to the use of the time step to control the execution speed of the simulation, the design of the simulation model allows for different representations of the network depending on traffic condition and level of congestion. During normal conditions, links are represented as single entities. Under queuing conditions, lanes are used to capture the correct spatial evolution of the queues and correctly model the spillback effects.

10.3.3 Online Estimation and Calibration of Parameters and Inputs

The online calibration component allows for the dynamic estimation of OD flows and other model parameters within the demand and supply simulators. Supply-side parameters include speed–density relationship parameters and output capacities of network links and intersections. Demand-side parameters (besides the OD flows) include behavioral model parameters.

Generally, the simulation models are calibrated off-line using a database of historic information. The calibrated parameter values are used in the online simulations. The calibrated model parameters represent average conditions over the period represented in the data. Such models are effective for off-line evaluation studies, which test the expected performance of traffic management strategies.

However, such models may not be effective for real-time applications, which analyze the system performance on the given day. If the off-line calibrated model is unadjusted, the predictive accuracy of the simulation system will decrease as it will not be sensitive to the variability of the parameters that affect traffic conditions between days.

DynaMIT can calibrate key parameters and inputs online. It starts from the off-line calibrated mean parameter values and steers them closer to the *realized* values. The most recent surveillance data is used to systematically re-calibrate the model parameters online, during every estimation interval.

Model formulation. In DynaMIT, the online calibration problem is the joint estimation of the unknown demand and supply parameters (including OD flows, route choice model parameters, traffic dynamics models parameters, and segment capacities) at time period h, using data observed from time periods 1 to h (and parameters estimated in prior intervals). The problem of the real-time updating of parameters on an interval by interval basis (using current information) lends to a state-space formulation of the problem. A state-space model is defined by the following:

- A state vector that succinctly captures the state of the system through a number of variables;
- Measurement equations that capture the mapping of the state vector to the measurements; and
- Transition equations that capture the evolution of the state vector over time.

The state vector is the minimal set of data that is sufficient to uniquely describe the dynamic behavior of the system at a time interval (the assumption of a *discrete*, stochastic, dynamic system is made). The state vector includes the parameters that require calibration during each time interval h. The state is defined by $\mathbf{x}_h$ and $\boldsymbol{\beta}_h$, which represent the deviations of the OD flow vector and key model parameters, respectively, from their best available estimates. This formulation is general and can easily incorporate different sets of parameters as needed.

Ashok and Ben-Akiva (1993, 2000) formulated the OD estimation and prediction as a *state-space* model using deviations (instead of the actual flows). This approach is generalized in the online calibration for DynaMIT's inputs and key parameters. Suppose that a priori model parameters were estimated from historical data for several previous days or even months. These parameters embody information about the relationships that affect trip-making decisions and demand patterns and their temporal and spatial evolution. It is desirable to incorporate as much historical information as possible. A straightforward approach is to use *deviations* of the OD flows and key parameters from best available estimates as state variables. This allows the formulation to indirectly account for all available a priori structural information.

Unobserved factors that are correlated over time (like weather conditions and unusual events) lead to correlations of deviations over time. These are reflected by the autoregressive process. More specifically, this process is characterized by a set of coefficients that describe the effects of the deviations from one time interval on the deviation during a subsequent time interval. These coefficients are determined offline using a linear regression model for each OD pair/parameter and for each time interval. Predicted deviations are obtained by applying this autoregressive model to the deviations estimated for the current time interval. This process is described in more detail in Section 10.5.

Let $\mathbf{x}_h$ and $\boldsymbol{\beta}_h$ denote the OD flows and key model parameters (respectively) for time interval h. Available information is associated with the unknown parameter values through direct and indirect measurement equations. A priori values of the model parameters $\mathbf{x}_h^a$ and $\boldsymbol{\beta}_h^a$ provide direct measurements of the unknown parameters $\mathbf{x}_h$ and $\boldsymbol{\beta}_h$ during interval h. In the form of deviations from historical values, the direct equations can be written as

$$\Delta\mathbf{x}_h^a = \Delta\mathbf{x}_h + \mathbf{v}_h \tag{10.4}$$

$$\Delta\boldsymbol{\beta}_h^a = \Delta\boldsymbol{\beta}_h + \mathbf{w}_h \tag{10.5}$$

where $\mathbf{v}_h$ and $\mathbf{w}_h$ are vectors of random error terms.

The indirect measurement equations are given by

$$\mathbf{M}_h = f\left(\mathbf{x}_{h-p}, \ldots, \mathbf{x}_{h-1}, \mathbf{x}_h, \boldsymbol{\beta}_{h-p}, \ldots, \boldsymbol{\beta}_{h-p}, \boldsymbol{\beta}_h; G_{h-p}, \ldots, G_{h-1}, G_h\right) + \mathbf{u}_h \quad (10.6)$$

where $\mathbf{M}_h$ is the data collected from the surveillance system (e.g., counts, speeds, travel times) during time interval h, *f(.)* is the corresponding output from DynaMIT for a given set of inputs (OD flows, parameters, network), G_h is the network for interval h, including descriptions of incidents and road closures, p is the number of intervals required to complete the longest trip in the network (therefore included in the augmented state), and $\mathbf{u}_h$ is a vector of random error terms. Generally, modeled trips last longer than one interval. Traffic conditions during previous intervals (and consequently by the model parameters used during these intervals) impact the simulated conditions.

Transition equations capture the evolution of the state vector over time. A typical formulation for the transition equation is an autoregressive process that relates the state during a given interval to a series of states from previous intervals, as follows (in deviations' form):

$$\begin{bmatrix} \Delta\mathbf{x}_{h+1} \\ \Delta\boldsymbol{\beta}_{h+1} \end{bmatrix} = \sum_{q=h-p}^{h} F_q^{h+1} \begin{bmatrix} \Delta\mathbf{x}_q \\ \Delta\boldsymbol{\beta}_q \end{bmatrix} + \eta_h \quad (10.7)$$

where F_q^{h+1} is a transition factor capturing the impact of the state vector during interval q on the state vector during interval $h+1$ and η_h is a vector of random error terms of appropriate dimension. The components of the state vector (e.g., OD flows, speed–density relationship parameters, capacities) represent distinct aspects with different characteristics. Each of these may evolve over time according to a distinct autoregressive process.

Solution approach. Most methodologies used to solve state-space models are based on the Kalman Filter algorithm (Kalman, 1960), which is based on a prediction-correction cycle. Initial estimates of the state vector and its covariance are obtained at each time interval by applying the transition equation to *predict* the state (and covariance) for the new interval (time update). The filter uses the available measurements to *correct* this prior estimate of the state vector and its covariance (measurement update). The outcome is a filtered state vector and updated covariance. The original OD estimation and prediction implementation within DynaMIT used a linearization of the problem, in which the OD flows were mapped to the link flows through an assignment matrix (that was obtained as an output from the supply simulator). This approach had several limitations. It was difficult to model the adjustment of other model parameters through a similar "assignment" matrix. For example, the relationship of traffic dynamics model parameters may not be easily mapped to the link flows. The introduction of other surveillance data, such as travel times, could not be easily mapped. Finally, the assignment matrix-based approach has fixed-point properties (similarly to the information generation

problem) and it resulted in another iterative loop between the demand and supply simulators.

The measurement equations for DynaMIT are non-linear and do not even have an analytical representation. Extensions of the original Kalman filter are employed as a solution. The Extended Kalman Filter (EKF) employs a *linearization* of the non-linear relationship to approximate the measurement equation with a first-order Taylor expansion (Chui and Chen, 1999). This algorithm can be applied to simulation-based systems that do not have an analytical representation by performing the linearization step using numerical derivatives, which require a large number of function evaluations (each implying a run of the simulator), making this technique potentially computationally expensive.

Various studies (e.g., Antoniou et al., 2007) indicate that the use of the Limiting Extended Kalman Filter (LimEKF) is a better option. The LimEKF is a special case of the EKF that offers considerable computational advantages. The most *computationally intensive* step in the EKF algorithm is the linearization of the measurement equation, as it requires the use of numerical derivatives. In real-time applications, it may be possible to replace the Kalman Gain matrix by a constant gain matrix, considerably decreasing the computation time. The *limiting* Kalman Filter replaces the Kalman gain matrix with its "limit," called the *limiting* (or stable) *Kalman gain matrix* (Chui and Chen, 1999).

The limiting Kalman gain matrix can be computed off-line. Several strategies can be developed to improve the performance of the limiting Kalman gain. For example, the EKF could run off-line, with each run producing a new gain matrix. These matrices can be used to update the limiting Kalman gain matrix. Another strategy is to only consider the last few gain matrices, using averaging.

10.3.4 Graphical User Interface (GUI)

DynaMIT has a Java-based graphical user interface that can be used for various tasks, including the generation and editing of input files (such as network files and incident files), control of the simulation engine, and visualization of the model outputs. The main components of the GUI are shown in Fig. 10.6:

- The network view shows a view of the network and allows for visualization by, e.g., type of link, link flow, speed, or density or other properties. The user can pan (move) around the network and zoom in and out to areas of interest. The visualization type can also be changed interactively.
- The time horizon bar shows the simulation time and the current estimation and prediction intervals and allows the user to move across this period, visualizing the time-dependent model estimation and prediction results.
- The controls allow the user to switch between editing and viewing modes and start or stop the simulation.

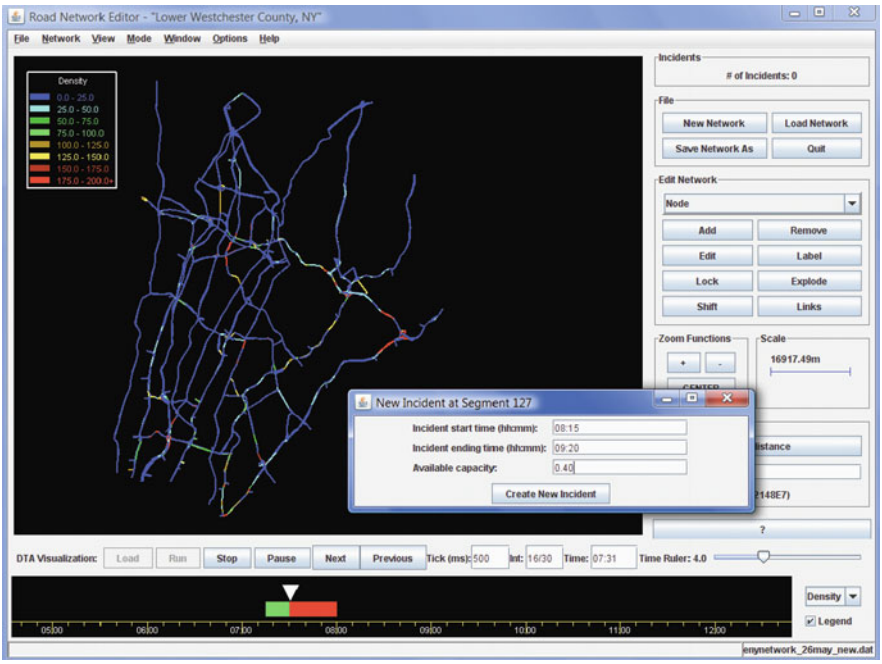


Fig. 10.6 DynaMIT GUI overview

10.4 Dynamic Traffic Assignment

DynaMIT combines the models presented in the previous section to estimate and predict the state of the traffic network and generate and disseminate traffic information.

Figure 10.7 outlines DynaMIT's state estimation process, which reproduces a full description of the current network conditions to match the available real-time data as closely as possible and estimates the key model parameters that will be used for state prediction and information generation. To perform state estimation, DynaMIT uses historical OD matrices, the travel time information provided to the travelers, and real-time traffic counts from the surveillance systems. Disaggregate behavioral models use the traffic information to update the departure time and route choice decisions of the travelers, which are then aggregated into updated OD matrices. The online calibration model in the state estimation uses the updated demand information and the latest available surveillance data to estimate OD flows and key model parameters (such as segment capacities). This information is loaded into the supply simulator, where the en route demand models are also used. The estimated network conditions and key estimated parameter values are produced and used as input for the prediction-based information generation.

The prediction-based information generation module of DynaMIT is shown in Fig. 10.8. While no measurements of the future traffic state exist, the objective is

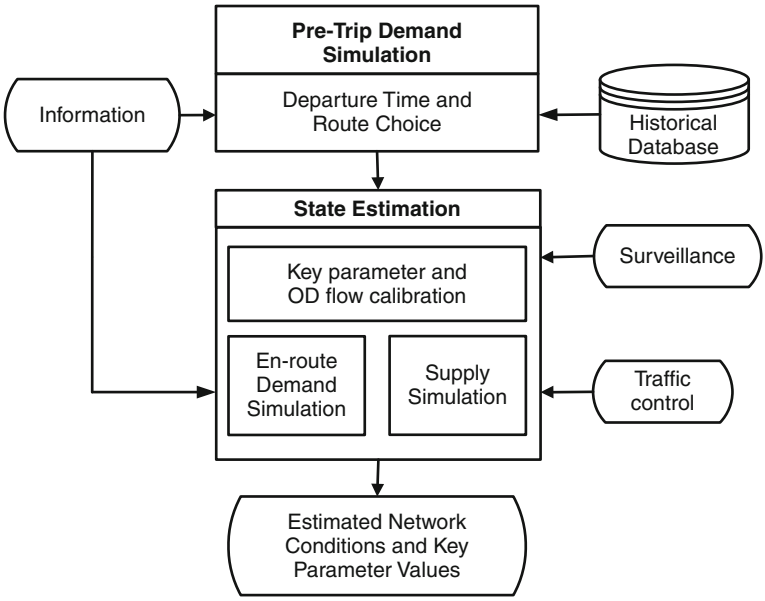


Fig. 10.7 State Estimation

to generate traveler information and guidance that is consistent with the traffic predictions. The procedure generating the traffic guidance depends on that very traffic guidance. This property gives fixed-point properties to the state prediction and information generation problem. The network state most recently estimated is used as a starting point for prediction. The most recently disseminated information and guidance is used as a trial strategy. The demand and supply simulators are used similarly for prediction as they are for estimation. The key difference is in the OD matrices and key model parameters that are used. DynaMIT relies on the autoregressive process described in Section 10.3.3 to forecast future OD matrices and key model parameters (e.g., speed–density model parameters and capacities). Based on the resulting prediction, a revised travel time-based information strategy is generated using a combination of the original and new predictions.

This revised strategy is in turn used by the coupled demand and supply simulators to produce a new prediction. Iterations continue until convergence. The developed algorithm is described in detail in Ben-Akiva et al. (1997) and Bottom et al. (1999).

10.5 Calibration and Validation

Off-line calibration is an essential step preceding DynaMIT deployment and creates a historical database that ensures DynaMIT's ability to replicate average or expected traffic conditions captured by archived sensor data over many days. The

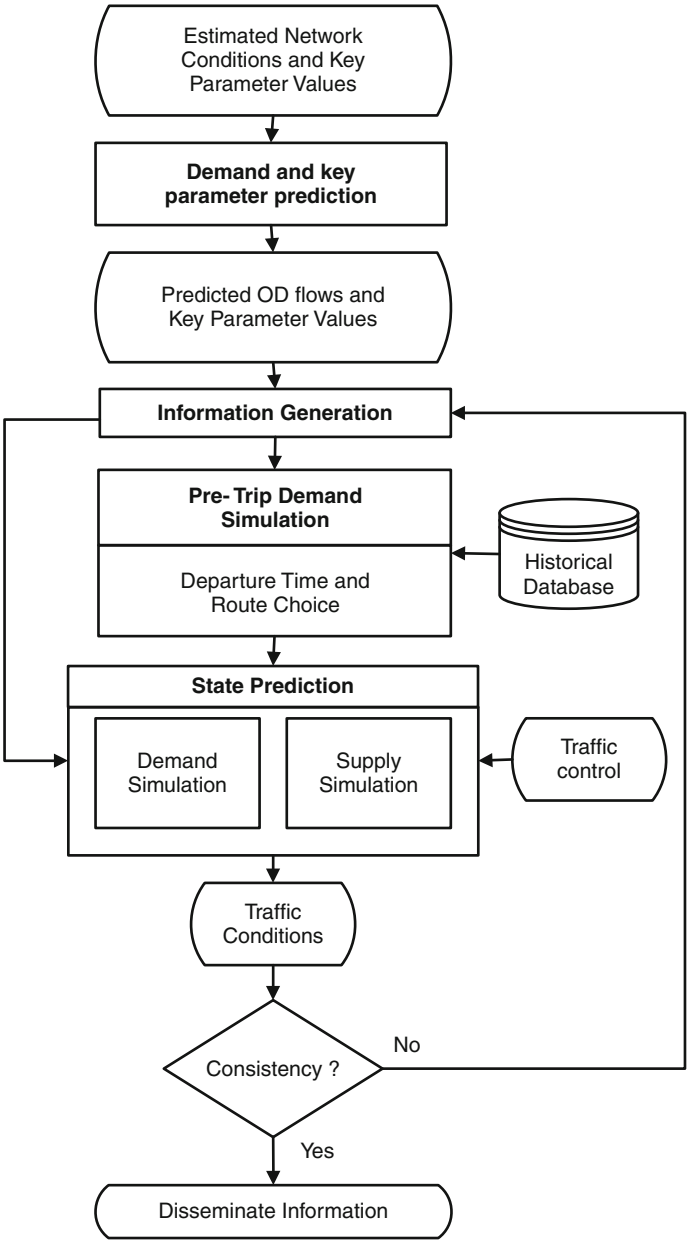


Fig. 10.8 Prediction-Based Information Generation

various inputs and parameters (such as OD flows, link performance functions, and route choice parameters) are estimated so as to replicate past traffic measurements. The demand and supply processes and their interactions that result in observed surveillance measurements could be affected by exogenous factors such as the day

of the week, time of day, precipitation levels, incidents, and the scheduling of special events or maintenance work zones. Off-line calibration must create a historical database containing a complete set of DynaMIT model inputs and parameters for every combination of factors observed in the data. Each set in such a stratified database should include estimates of time-dependent OD flows, capacities, link or segment performance functions, route choice model parameters, habitual network travel times, and other parameters used in estimating and predicting OD flows. Archived surveillance data from several days must be available for off-line calibration, which is a global optimization step aimed to determine the mean values of the model parameters and inputs.

10.5.1 Off-Line Calibration

Formulation. The analysis period is divided into uniform intervals $h = 1, 2, \ldots, H$. Let $\mathbf{M}_h$ denote the vector of available aggregate sensor measurements for interval h and $\mathcal{M}_h$ the corresponding estimate from DynaMIT. Let $\mathbf{x}_h$ be the vector of OD flows departing in interval h; $\mathbf{x}_h^a$ the initial OD flow estimates; and $\boldsymbol{\beta}_h$ a vector of route choice and supply model parameters; $\boldsymbol{\beta}_h^a$ the corresponding initial values. The off-line calibration problem can be formulated to minimize an objective function z:

$$(\boldsymbol{\beta}^*, \mathbf{x}^*) = \arg\min_{\boldsymbol{\beta},\mathbf{x}} \; z = \sum_{h=1}^{H} \left[z_1 \left(\mathbf{M}_h, M_h \right) + z_2 \left(\mathbf{x}_h, \mathbf{x}_h^a \right) + z_3 \left(\boldsymbol{\beta}_h, \boldsymbol{\beta}_h^a \right) \right] \tag{10.8}$$

subject to

$$\mathcal{M}_h = f \left(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_h, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_h; G_1, G_2, \ldots, G_h \right) \quad \forall h \tag{10.9}$$

$$\lambda_h^x \leq \mathbf{x}_h \leq \upsilon_h^x \quad \forall h \tag{10.10}$$

$$\lambda_h^\beta \leq \boldsymbol{\beta}_h \leq \upsilon_h^\beta \quad \forall h \tag{10.11}$$

where $f(\cdot)$ denotes DynaMIT output; $\mathbf{x} = \{\mathbf{x}_h | h\}$ and $\boldsymbol{\beta} = \{\boldsymbol{\beta}_h | h\}$. The optimal estimates are denoted by $\mathbf{x}^*$ and $\boldsymbol{\beta}^*$. $G = \{G_h | h\}$, with G_h representing the network for interval h, including descriptions of incidents and road closures. The elements of G are not variables in the optimization problem, though their values impact the output from DynaMIT. Equations (10.10) and (10.11) represent lower (λ) and upper (υ) bounds on the OD flows and model parameters, respectively.

In the short term, network travel times are assumed to be in a steady state. The day-to-day evolution of travel times is likely to be minimal for the available sensor data owing to the high fraction of urban commuters familiar with the network. Relaxation of these assumptions is possible (Balakrishna, 2006).

The goodness-of-fit measures z_1, z_2, and z_3 capture the error between the simulated or estimated quantities and their measured or a priori values. These functions

must be selected based on a priori hypotheses regarding the distributions of the appropriate error terms**.** The most common assumption is that of normally distributed errors, which results in a generalized least squares (GLS) formulation (Cascetta et al., 1993):

$$(\boldsymbol{\beta}^*, \mathbf{x}^*) = \arg\min_{\boldsymbol{\beta},\mathbf{x}} z = \sum_{h=1}^{H} \Big[(\mathbf{M}_h - \mathcal{M}_h)^T \mathbf{U}_h^{-1} (\mathbf{M}_h - M_h) + (\mathbf{x}_h^a - \mathbf{x}_h)^T \mathbf{V}_h^{-1} (\mathbf{x}_h^a - \mathbf{x}_h) + (\boldsymbol{\beta}_h^a - \boldsymbol{\beta}_h)^T \mathbf{W}_h^{-1} (\boldsymbol{\beta}_h^a - \boldsymbol{\beta}_h) \Big] \tag{10.12}$$

$\mathbf{U}_h$, $\mathbf{V}_h$, and $\mathbf{W}_h$ represent the variance–covariance matrices of the measurement errors associated with $\mathbf{M}_h$, $\mathbf{x}_h^a$, and $\boldsymbol{\beta}_h^a$, respectively. They may be used to capture any correlations among the various quantities. For example, the elements of $\mathbf{U}_h$ represent the potential spatial and temporal dependencies between sensor count and speed measurements: Counts and speeds at a point on the network may be correlated, while counts (or speeds) on a link may be correlated with those measured immediately upstream of that link. If the variance–covariance matrices are assumed to be diagonal, the reciprocals of the non-zero elements may be interpreted as relative weights on the various observed or a priori quantities.

Solution approach. The solution of the off-line calibration methodology requires an appropriate algorithm that recognizes the special characteristics of the problem. The algorithm must be appropriate for problems with a highly non-linear, noisy objective function and capable of moving past local optima toward a global solution. The algorithm must also minimize the number of costly model runs required for objective function evaluation.

Stochastic approximation (SA) approaches are good candidates for the solution of such problems. The SA algorithm, Finite Difference SA (FDSA), is conceptually analogous to numerical derivation. FDSA requires a large number of function evaluations, which is computationally expensive. The number of function evaluations is $2n$, where n is the number of parameters and inputs to be calibrated.

Simultaneous Perturbation Stochastic Approximation (SPSA) (Spall, 1998) is a similar approach that approximates the derivative by perturbing all parameters simultaneously and only requires a fixed number of function evaluations (two) irrespective of the problem dimensions. The SPSA algorithm showed high-quality solutions with computational efficiency in recent experiments (Balakrishna, 2006; Balakrishna et al., 2007).

The solution of the off-line calibration problem requires knowledge of the variance–covariance matrices $\mathbf{U}_h$, $\mathbf{V}_h$, and $\mathbf{W}_h$. If these are not known a priori, their elements can be estimated using a Feasible Generalized Least Squares (FGLS) procedure (Greene, 2000). FGLS is an iterative approach where the variance–covariance matrices are initialized to identity matrices of appropriate dimensions, and the DTA model is calibrated accordingly. The residuals $(\mathbf{M}_h - \mathcal{M}_h)$, $(\mathbf{x}_h^a - \mathbf{x}_h)$, and $(\boldsymbol{\beta}_h^a - \boldsymbol{\beta}_h)$ from this Ordinary Least Squares (OLS) step are then used to approximate the variance and covariance. The calibration step can be repeated until stable variance–covariance estimates are obtained.

Building a historical database. The process of creating a historical database requires time-dependent traffic data from several days so that all relevant real-world demand and supply patterns are covered in the database. Balakrishna et al. (2005) provide a methodology to estimate historical inputs and parameters on the demand side using traffic count data from M days, assuming that the supply parameters can be estimated independently.

The archived surveillance data set contains observations spanning M days. A static OD matrix is assumed to be available, potentially from a planning model. The variables of interest (whose values will be stored in the historical database) are the route choice model parameters, time-dependent historical travel times for each interval, historical OD flows, OD prediction coefficients, and error covariance matrices for the various measurement equations.

Without loss of generality, a single-day type consisting of M days of observations is selected to illustrate the approach. It is assumed that the sensor data for all M days were generated by the same underlying demand patterns and mean OD flows. A common time-dependent habitual travel time profile is also assumed. The database of historical information and model parameters is developed through a heuristic procedure by sequentially processing the M days of data to deal with the complexity and large-scale nature of the problem. The historical database is thus updated incrementally after each day.

The OD flows and error covariance matrices for day m are estimated using FGLS, and the process is repeated daily so that the outputs generated at the end of day m form inputs for day $m+1$. OD prediction coefficients such as the autoregressive matrices (see eq. (10.13)) are assumed to be relatively constant across days and estimated periodically every k days. These matrices may be estimated via standard regressions. The parameters of the route choice model may also be estimated periodically. The route choice parameter estimates can be determined by solving an optimization problem that minimizes some measure between the simulated and observed counts, given a priori parameter estimates as constraints. Habitual travel times for a given set of route choice model parameters and time-dependent OD matrices are then obtained by an iterative weighting process such as the method of successive averages (MSA) (Cascetta et al., 2002).

10.5.2 Validation

In general, the validation of advanced models, such as DynaMIT, can take place according to the dimensions illustrated in Table 10.1.

Evaluation can take place in the laboratory (off-line) or the field (online). An online test uses real-time traffic data directly from the surveillance system. An off-line test replaces the online communication with an archived data set or a real-world simulator such as MITSIMLab (see Chapter 6). The evaluation can also be *open loop* or *closed loop*. An open loop evaluation compares estimated and predicted network performance measures (which form the basis for guidance generation) with

Table 10.1 DynaMIT validation dimensions

	Open loop	Closed loop
Off-line	Archived data, simulation laboratory, NO information dissemination (Irvine, Los Angeles, Southampton)	Simulation laboratory, information dissemination (Lower Westchester County-LWC)
Online	Real-time data feed, NO information dissemination (Los Angeles)	Real-time data feed, information dissemination (no application yet)

corresponding real world traffic measurements. However, guidance is not disseminated to the drivers. A closed-loop evaluation provides the feedback mechanism to inform equipped drivers of the current guidance strategy and evaluates the drivers' responses to the disseminated information.

The above classification results in four possible evaluation approaches:

- *Off-line, open- loop evaluation*. Archived surveillance data is used to estimate the current network state. State predictions for future time periods are compared to the corresponding observations in the archived data set.
- *Off-line, closed-loop test*. A simulation laboratory replaces the real world. Traffic data from the simulated surveillance system is transmitted in real time to DynaMIT. DynaMIT generates predictive guidance and sends it to equipped drivers in the microscopic simulator, and network performance measures are computed to ascertain the effectiveness of the guidance.
- *Online, open-loop test*. Traffic data received in real time from the surveillance system is used to perform state estimation, state prediction, and guidance generation. The predicted network state is compared with sensor measurements as they become available to validate the congestion reduction capability of the guidance generation system.
- *Online, closed-loop test*. The feedback loop is closed, with real, equipped drivers on the network receiving the generated guidance.

The estimation and prediction capabilities of DynaMIT have been evaluated in various case studies (mentioned in Table 10.1). In the sections that follow, we report on results of two applications that represent the case of off-line open-loop evaluation (Los Angeles and Southampton, UK). The Los Angeles case results from both off-line and online, open-loop evaluation are also presented. Further evaluation results are shown within the applications presented in Section 10.7.

Los Angeles, CA. The Los Angeles case study applies the calibration methodology described in earlier sections and demonstrates the validity of the model. The network consists of 243 nodes and 606 links (Fig. 10.9). Archived freeway and arterial flow and speed data from 203 loop detectors were obtained on September 2004 from PeMS (UC Berkeley and Caltrans, 2005) and LA DoT, respectively. Other surveillance data included an incident log and weather conditions in the area.

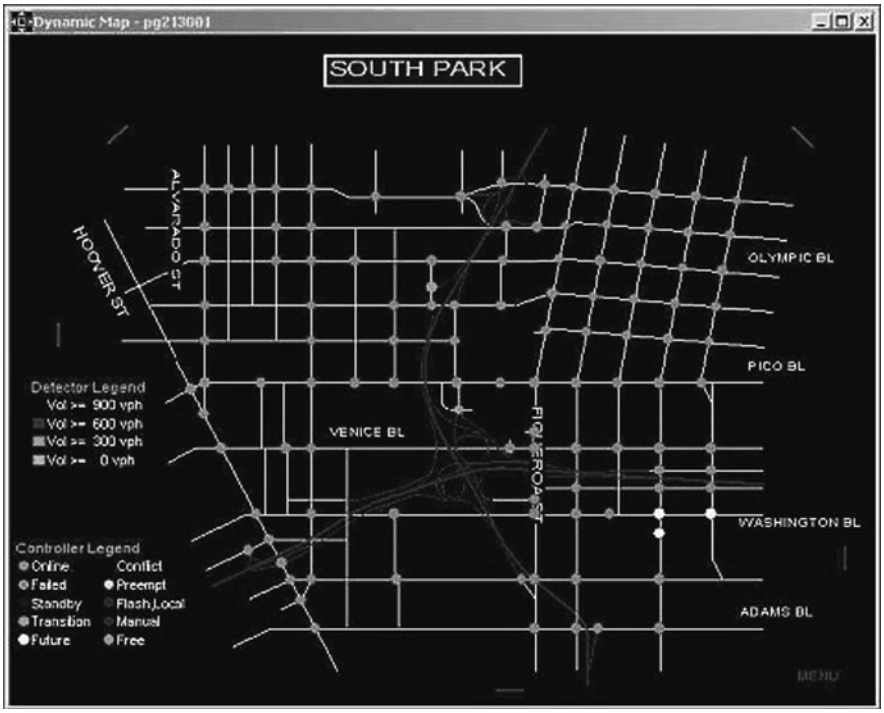


Fig. 10.9 South Park, Los Angeles Network

Weekday data were selected for this case study. The 3:00–9:00 a.m. period (divided into 15-min intervals) was chosen to capture the a.m. peak. The network was fairly congested with a large number of route choices for each of the 1129 OD pairs. Time-dependent OD flows were calibrated for each interval, together with route choice model parameters, segment capacities, and speed–density function parameters.

The performance of DynaMIT was assessed using the normalized root mean square error (RMSN)

$$\text{RMSN} = \frac{\sqrt{N \sum_N \left(y - \hat{y}\right)^2}}{\sum_N y} \tag{10.13}$$

where N is the number of measurements, y denotes the observed, and $\hat{y}$ the estimated traffic parameters, e.g., counts or speeds. Speeds are difficult to estimate and represent a challenging test for the proposed approach (detailed results of the case study are given in Balakrishna et al., 2007).

The calibration was performed using the methodology discussed in Section 10.5.1. The off-line calibration results are summarized in Table 10.2. "D" corresponds to simultaneous demand calibration across all intervals. "SD" denotes

Table 10.2 Fit to counts and speeds, RMSN (off-line calibration)

	Fit to counts		Fit to speeds	
Estimator	Freeway	Arterial	Freeway	Arterial
Reference	0.218	0.239	0.181	0.203
D	0.114	0.143	0.118	0.125
SD	0.090	0.113	0.088	0.093

simultaneous demand and supply calibration. The reference case refers to the traditional approach that calibrates the demand and supply parameters sequentially and independently. In addition, the demand estimation in the reference case takes place sequentially from interval to interval. The statistics demonstrate that the calibration methodology provides satisfactory fit to count and speed observations. The numerical results are similar for freeways and arterials, indicating the methodology's ability to accurately capture spatial and temporal dynamics. Figure 10.10 shows the comparison of cumulative sensor counts at a sample location on the network and indicates the improvement in model accuracy when compared with the reference case.

Open-loop tests were used to assess DynaMIT's traffic prediction ability where DynaMIT estimated and predicted traffic conditions (link flows) in a rolling horizon mode. Figure 10.11 illustrates results for time interval 8:30–8:45 a.m. for estimation and prediction. Prediction results include one-step (15 min ahead) and two-step

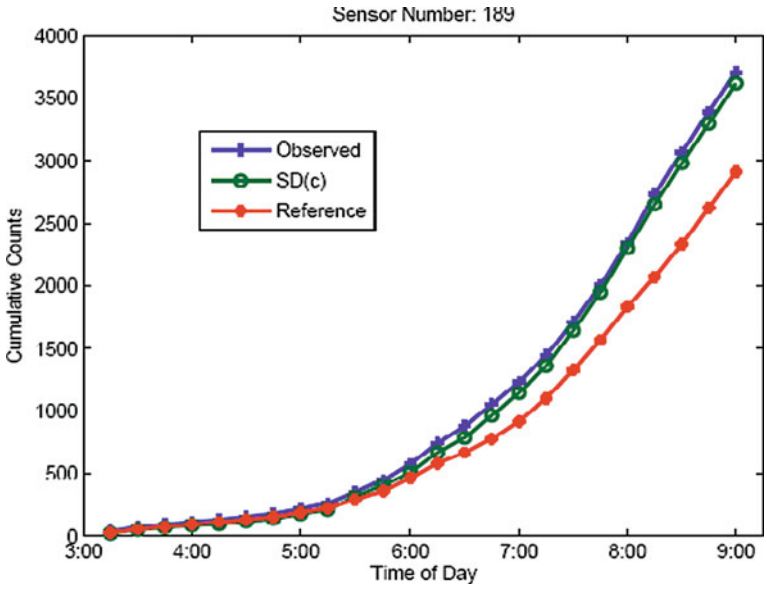


Fig. 10.10 Comparison of Cumulative Counts

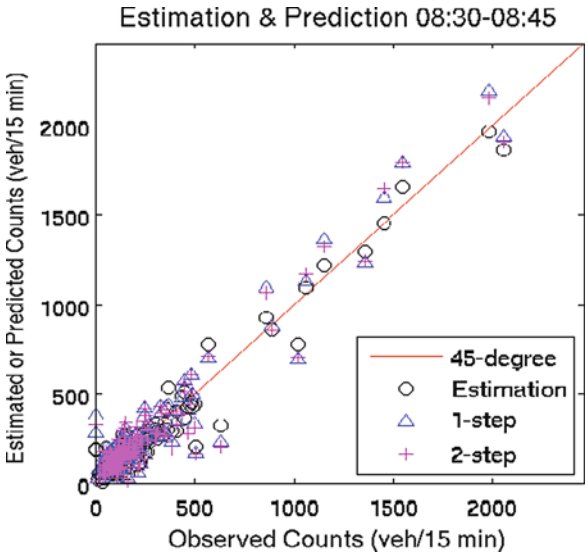


Fig. 10.11 Validation: Estimation and prediction results (counts)

predictions (30 min ahead). The results in Fig. 10.11 verify DynaMIT's ability to estimate and predict actual traffic conditions (Wen et al., 2007).

Southampton. DynaMIT was validated in Southampton, UK (Fig. 10.12), in an off-line, open-loop mode. Surveillance data from ten sensor locations (seven on the mainline and three on on-ramps) over 5 weekdays were used for calibration. Available surveillance data included counts, speeds, and densities. The estimated parameters include flows from 20 OD pairs, link capacities from 45 link segments, and speed–density relationship parameters. Regarding speed–density parameters, the segments were divided into three groups (two for the main-line segments and one for the ramp segments) for a total of 15 parameters to be calibrated.

An off-line calibration provides the starting values for the online calibration. The RMSN (normalized root means square error) statistic for the off-line calibrated

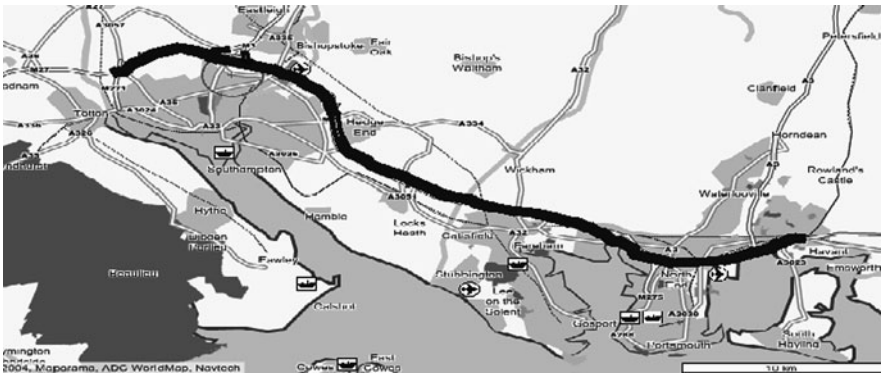


Fig. 10.12 Southampton network (source: Maporama)

counts for this network was 0.1232. The total fit of the off-line calibrated speeds, as quantified by the RMSN, was 0.1102.

The online calibration methodology was applied between 12:00PM and 6:00PM on a day not used for off-line calibration. Both EKF and LimEKF solution algorithms were used for comparative purposes. The performance of the two solution algorithms and the output of the online calibration are summarized in Table 10.3, using RMSN as measure of effectiveness.

Table 10.3 Fit to counts and speeds

	Speeds		
	Estimated	One-step predicted	Two-step predicted
Online (EKF)	0.1106	0.1209	0.1303
Online (LimEKF)	0.1120	0.1249	0.1347
	Counts		
	Estimated	One-step predicted	Two-step predicted
Online (EKF)	0.1039	0.1318	0.1550
Online (LimEKF)	0.1091	0.1321	0.1702

The Limiting EKF is the best candidate for the wide deployment of such approaches despite the slightly higher estimation and prediction accuracy of the EKF algorithm. The LimEKF algorithm performed comparably to the EKF and is significantly less computationally complex. Furthermore, the LimEKF algorithm is scalable as its computational complexity (i.e., number of function evaluations needed) does not depend on the problem size.

Figure 10.13 demonstrates the impacts of online calibration in adjusting the speed–density relationships. The off-line calibrated speed–density relationship is

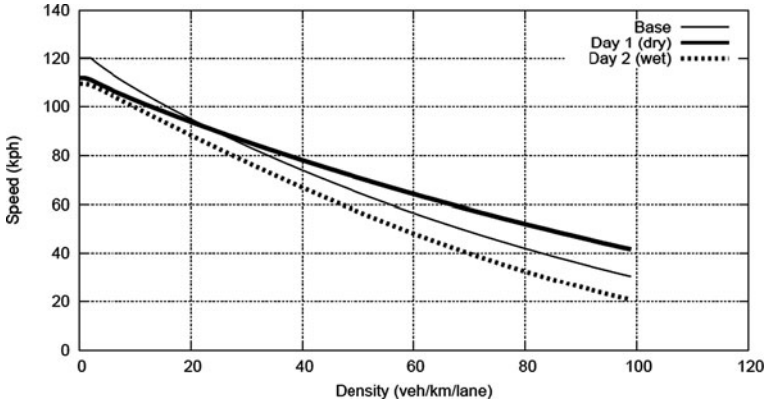


Fig. 10.13 Online calibrated speed–density relationships

indicated by the dashed curve. This relationship captures average traffic conditions and was estimated using data from several days. The online calibration for a particular day adjusts the off-line calibrated speed–density relationship to capture prevailing conditions for the day and time interval. For example, during wet weather conditions, drivers use lower speeds for the same density. Therefore, the speed/density observations fall below the average speed–density curve (dashed line). The online calibrated speed–density curves for dry weather exhibit the opposite behavior.

10.6 Extended Modeling Capabilities: Working with External Applications

10.6.1 Interface with Other TMC Applications

At TMCs, DynaMIT is expected to interact with other applications such as incident detection or signal control applications. DynaMIT was deployed at the ATSAC (Automatic Traffic Surveillance and Control) TMC in Los Angeles (Fig. 10.14), where it obtained surveillance and incident information from the surveillance system to perform traffic state estimation and generate predictions. These functionalities were evaluated online. The estimations and predictions were fed to the TMC infrastructure for future use by incident detection algorithms (such as CLAIRE and

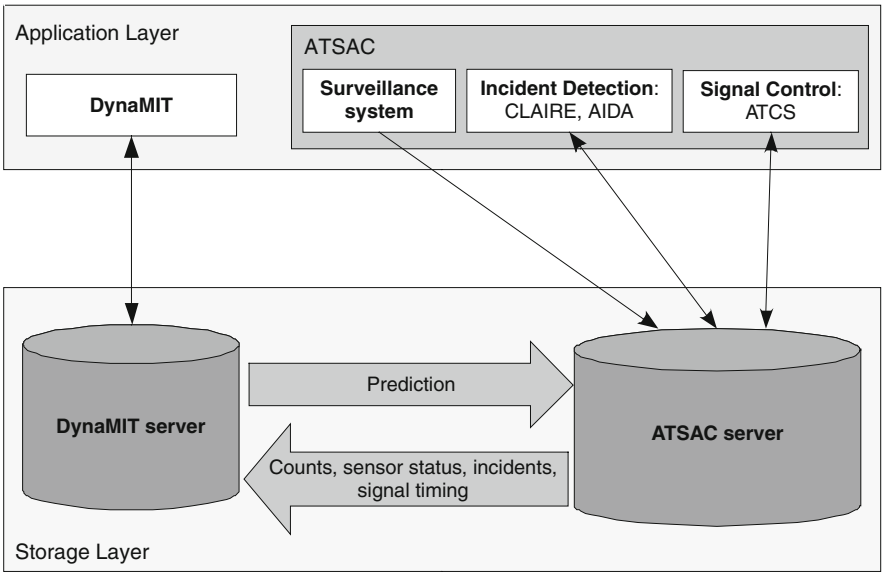


Fig. 10.14 DynaMIT deployment at the ATSAC TMC

Advanced Incident Detection Algorithm, AIDA) and signal control systems (such as ATSC). For more information on this application, see Wen et al. (2007).

10.6.2 Closed Loop

DynaMIT requires large amounts of traffic surveillance data and incident information to generate accurate outputs. For real-time systems, such outputs should be distributed back to users. The developed software interfaces link DynaMIT with the TMC (and with microscopic traffic simulation models, such as MITSIMLab, when they act as a proxy to the TMC, as presented in Chapter 6) in a closed-loop operation, providing a streamlined facility for real-time information exchange.

The closed-loop operation allows information to flow continuously, without external interference. The information includes surveillance information (including loop detector counts and TRANSMIT measurements), accident reports (location, duration, severity), and traffic messages disseminated to the drivers (e.g., through Variable Message Signs, VMS, or Highway Advisory Radio, HAR). DynaMIT uses this information to estimate and predict traffic conditions and generate guidance. The outputs are transmitted back to the TMC to generate new traffic management strategies, thus affecting the surveillance that will be collected and transmitted to DynaMIT.

10.6.3 Innovative User Interfaces Through Mash-ups/Web Services

A prototype web-based GUI is in development, as shown in Fig. 10.15. It converts DynaMIT's output files to an XML format and leverages Google Maps services for visualization. The GUI overlays the speed, density, or flow information on the digital maps or satellite images. A time line is inserted in the digital map, allowing a minute-by-minute view of network conditions. The bottom panel shows the speed profile of selected sections in the network.

10.7 Selected Overview of Advanced Case Studies and Applications

10.7.1 Irvine: Predictive VMS

In Irvine, CA, DynaMIT generated predictive guidance during incidents, through variable message signs (VMS) (see Sundaram, 2002, for details). The network consists of 298 nodes and 618 freeway and arterial links (Fig. 10.16). A base case representing non-incident (congested) conditions was developed using data from the peak period between 7:15 and 8:30 a.m.

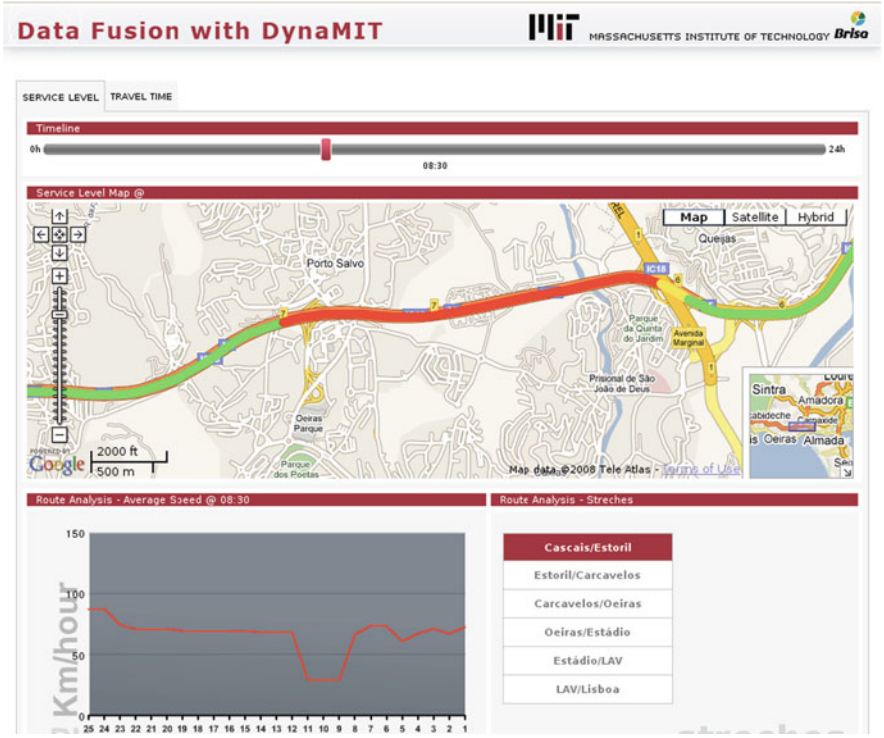


Fig. 10.15 Next generation DynaMIT GUI based on Google Maps

To calibrate DynaMIT, counts and speeds from 68 standard loop detectors were used. Five weekdays of traffic sensor count data from 68 sensors were available. A static planning matrix of the flows from 655 OD pairs, covering the morning peak period (7:00–10:00 a.m.) was also available. Figures 10.17 and 10.18 illustrate the calibrated DynaMIT's ability to estimate observed conditions. Figure 10.17 (left) demonstrates the accuracy of estimated counts during the interval between 7:30 and 7:45 a.m. across all sensors in the network. Figure 10.17 (right) compares the estimated and observed flows for one arterial sensor.

The fit to speeds achieved with the calibrated DynaMIT is illustrated in Fig. 10.18. The left figure shows the calibrated DynaMIT speeds. The right figure shows the validated speeds.

The above base case for the VMS analysis was perturbed with a hypothetical incident that removed 60% of the segment capacity from 7:17 to 7:40 a.m. Drivers departing between 7:15 and 8:15 experienced an average travel time of 1455 s (or 24.25 min). Nine hundred and seventy-two fewer travelers completed their trips in the same interval. The travel time for the primary OD pair, affected by the incident, increased by 494 s. The majority of travelers experienced travel times between 1500 and 2000 s as opposed to 1000–1500 s under incident-free conditions.

Fig. 10.16 Irvine, California Network

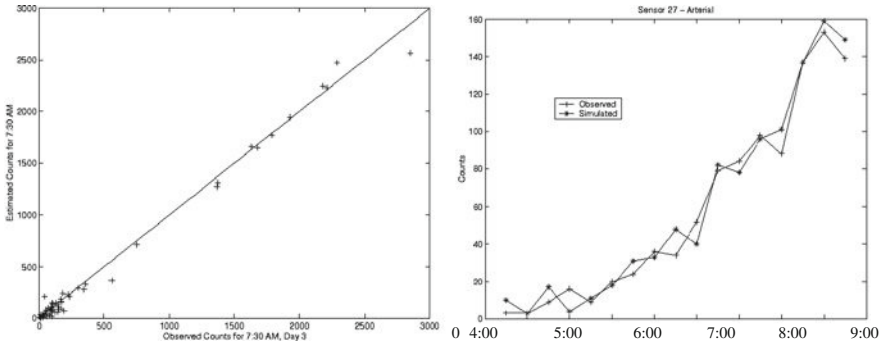


Fig. 10.17 Estimated Counts

When predictive information generated by DynaMIT was broadcasted through the VMS, the average travel time for those departing between 7:15 and 8:15 a.m. was reduced to 1401 s. One hundred and fifty-six more vehicles completed their trips during the same interval. Figure 10.19 shows the number of travelers in each travel time range for three scenarios: base case with normal traffic conditions, incident without any information provision, and incident with provision of predictive guidance. When the DynaMIT-generated predictive guidance was disseminated,

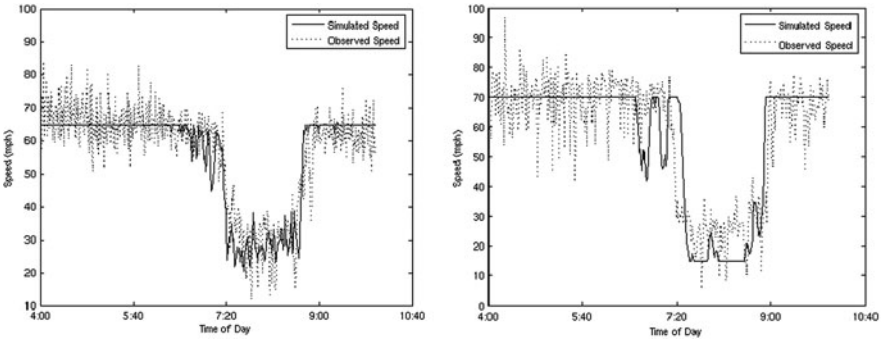


Fig. 10.18 Calibrated and validated DynaMIT speed profiles

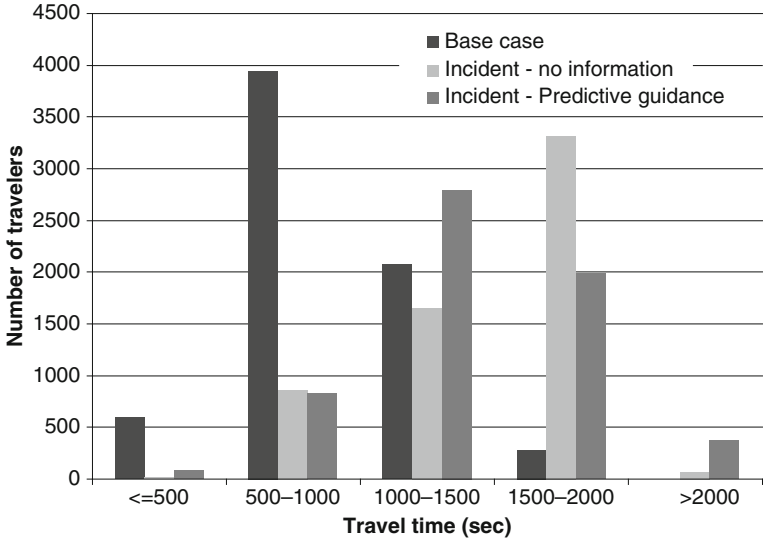


Fig. 10.19 Travel time frequency per scenario

travel time decreased substantially. These results illustrate the benefits of predictive guidance at a network level.

10.7.2 Lower Westchester County, NY – Incident Diversion Strategies

The objective of this case study was to generate and evaluate diversion strategies and to evaluate the effectiveness of their dissemination through VMS. The impact of data fusion into the predictive capabilities of DynaMIT was also tested using data from conventional loop detectors and automatic vehicle identification (AVI) sensors.

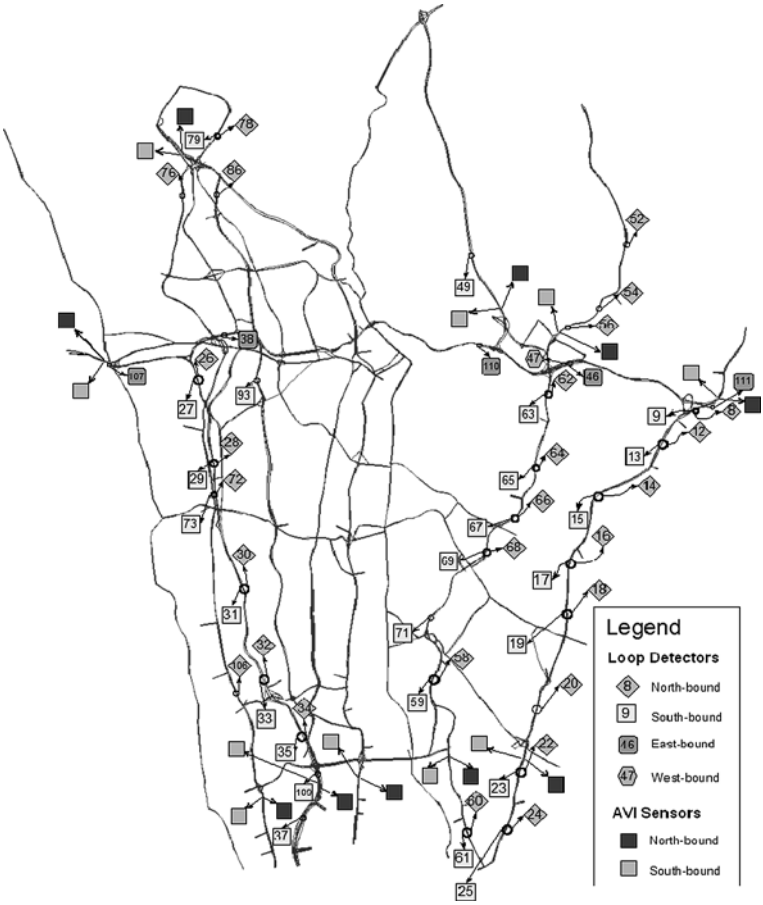


Fig. 10.20 Lower Westchester County network

The case study was implemented off-line using the closed-loop facilities outlined in Section 10.6.2. The network covers the Lower Westchester County (LWC) and includes important highway corridors (Fig. 10.20).

This case study demonstrates adversities that may occur when building dynamic models in real applications. The available data was limited and of poor quality (i.e., data from neighboring sensors were collected during different time periods, resulting in inconsistencies in the traffic counts). The study network contains freeways, parkways, and arterials. Due to restrictions on the parkways (i.e., no heavy commercial vehicles, such as trucks or trailers, are allowed to enter these facilities), commercial vehicle proportions were approximated based on data from three toll plazas.

There were 6470 parameters that were calibrated, including OD flows, segments capacities, and speed–density relationship parameters. The SPSA solution

Table 10.4 Calibration and validation results for Lower Westchester County network

	Demand-only calibration		Joint demand-supply calibration		Joint demand–supply validation
RMSN	Link counts (%)	Counts + AVI (%)	Link counts (%)	Counts + AVI (%)	Counts + AVI (%)
Counts	20.09	18.24	20.69	17.94	17.63
Travel Times	22.23	21.24	19.46	18.92	18.55

approach was selected for this case study. A summary of the key calibration and validation results is shown in Table 10.4. The calibration of demand and supply parameters improves the fit of the model. Combining link counts with AVI information improves the calibration results. The validation suggests that the calibrated parameters were not over-fit to the calibration data.

Based on major diversion routes and the most frequent incident locations provided by the NYSDOT, two typical incidents were simulated and route guidance was generated and evaluated. Each incident was modeled separately to identify its impacts. The incidents were located on Sprain Brook Parkway and Interstate-95 southbound, affecting the morning commute. The incident durations were 1 h, occurring at 7 a.m and cleared at 8 a.m. They blocked all lanes in the southbound direction. This level of incident severity is common in this network. A minor incident lasts for 30–60 min. Major incidents may last 3–4 h, during which all lanes are blocked. The incident caused average travel times between 6:30 and 10:30 a.m. (time interval that is affected by the incident) to increase by more than 80% in some cases.

Figure 10.21 presents a sample of the results. The comparison reveals that a significant number of vehicles with lower travel times (500–1500 s) shifted to higher travel times (> 2500 s). The impact of the route guidance is obvious since there is a reduction in the number of people with travel times greater than 4500 s due to predictive route guidance provided through the appropriate VMS. For more details, see Rathi et al. (2008).

10.7.3 Other Applications

DynaMIT has been further developed and used for applications related to emergency evacuation management, congestion pricing, and data fusion for a number of diverse data sources.

Emergency evacuation management. DynaMIT can be used off-line to generate potential network management strategies during emergency evacuations and online to refine and adjust evacuation plans in response to developing conditions. DynaMIT was used to evaluate evacuation strategies for the greater Boston Metropolitan area

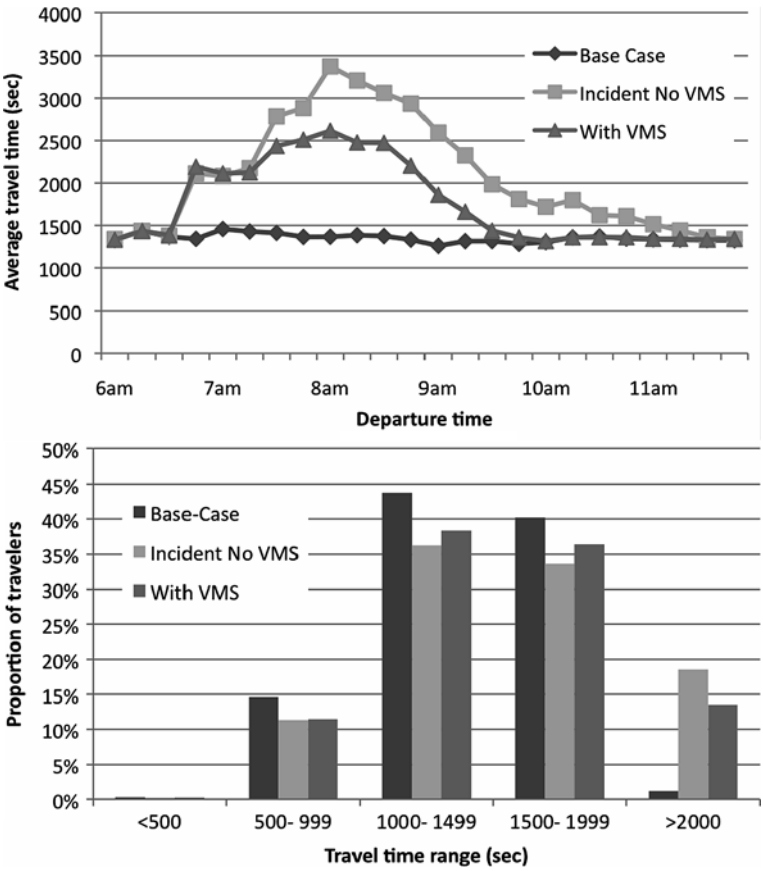


Fig. 10.21 Average travel time by departure time (top); travel time distribution (bottom)

(Balakrishna et al., 2008). The case study used contra-flow as a capacity-augmenting strategy.

Dynamic congestion pricing. DynaMIT was used to evaluate dynamic congestion pricing schemes. Congestion pricing aims to control network traffic by charging tolls. Typically, travelers will react by changing their behavior, such as diverting from the tolled roads, switching transportation modes, adjusting departure time, or even canceling their trips (Xu, 2009). DynaMIT incorporates travelers' route choices and departure time choices in response to congestion pricing.

Data fusion. DynaMIT is suited for data fusion applications where multiple data sources are available and a single predicted network state is sought. This capability was explored in two research projects: (a) CityMotion of the MIT-Portugal program and (b) traffic applications in Beijing, China. CityMotion's objectives are to obtain historical and real-time data from transportation-related resources in a Portuguese city in order to perform data fusion and provide coherent access to the

data. CityMotion also aims to build a pilot service to illustrate the use of integrated data to citizens for trip planning and to policy makers as a decision support tool. In this project, DynaMIT is using data from sensors (speeds, counts, occupancies), travel times, etc. The Beijing application fuses historical data sets that include static demand during the a.m. peak hours for 2927 OD pairs, counts and speeds from 134 Remote Traffic Microwave Sensors, and travel times from Floating Car Data. The Beijing application is challenging because of the urban congestion and network size. Approximately 600,000 vehicles were simulated for the a.m. peak period (6–10 a.m.).

10.8 Modeling Details of Advanced Case Studies

DynaMIT meets the computational challenges of a decision support system at a TMC, which requires real-time traffic predictions. Such applications can involve large networks (like in Beijing). In this section, we discuss extensions of the model to allow for distributed implementation. Using parallel simulation on a cluster of personal computers (PCs) is a cost-effective way to enhance the scalability of DynaMIT for large-scale online applications (Wen, 2009).

The strategy is based on a decomposition of the network to improve the computational efficiency of the traffic simulations. The main goal is to design partitioning strategies that achieve load balancing and minimize interprocessor communications. However, in the applications targeted by DynaMIT, traffic conditions are dynamic. The decomposition strategy is adaptive and adjusts to changing network conditions due to increased demand during certain time periods, incidents, etc. The strategy is implemented in a way analogous to the off-line and online calibration of the parameters. Some partitions are generated off-line for various time periods, based on historical data. The partitions may then be adjusted online to accommodate changing traffic conditions, as Fig. 10.22 illustrates.

The parallel version of DynaMIT, DynaMIT-MPI, uses distributed memory architecture. This scales better than shared memory systems. MPICH2 (Argonne National Laboratory, 2008), a high-performance and widely portable implementation of the MPI standard (Argonne National Laboratory, 2008), is the underlying library for communication. DynaMIT-MPI implements the adaptive network decomposition framework. It collects running time statistics and the spatial distribution of vehicles in the network and uses them to facilitate the generation of off-line partitions. During online applications, it loads suitable off-line partitions and decides whether or not re-partitioning is necessary. Another important feature of DynaMIT-MPI is its synchronization-feedback design, which is used to efficiently ensure the correctness of the boundary traffic dynamics.

DynaMIT-MPI was tested on two networks in NY and LA. Both used the same data sets from previous sections in this chapter. The Lower Westchester County, NY, network (Fig. 10.20) contains 825 nodes, 1767 links, and 482 OD pairs. Three processors, including two Intel Xeon CPUs at 3.00 GHz and one Intel Pentium 4

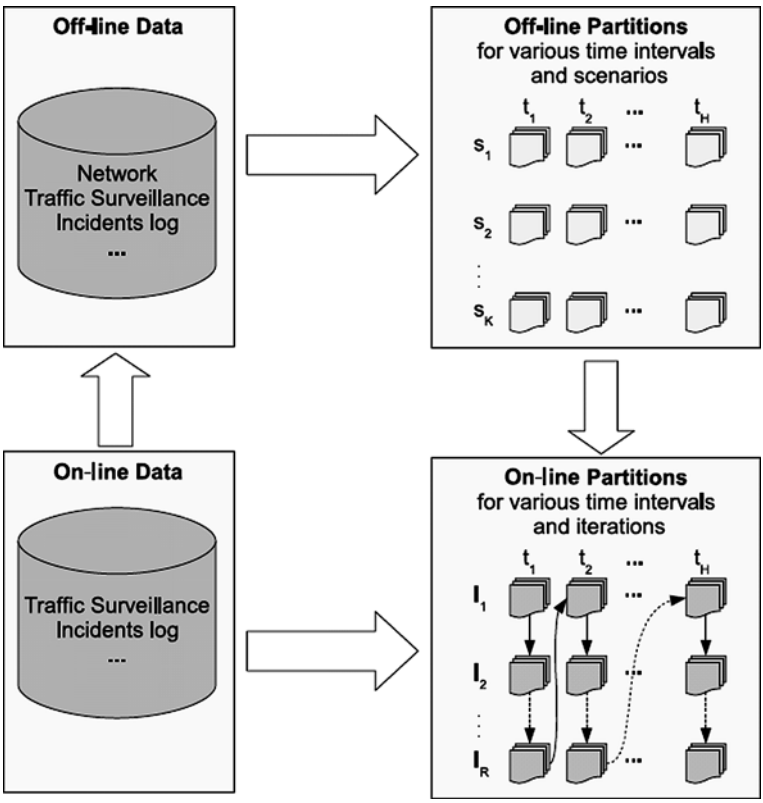


Fig. 10.22 Adaptive network decomposition framework

CPU at 3.60 GHz, were used. The Pentium 4 runs about 1.16 times faster than the others. All three computers were connected to a 100 Mbps LAN.

DynaMIT-MPI generates results comparable to the sequential version of DynaMIT. Good speed-up ratios were also found. Figure 10.23 shows the effectiveness of the parallelization approach, especially the adaptive strategy. An incident is generated that creates a lot of congestion. "BC" stands for base case, with a normalized running time of 100%. "MP" and "MA" refer to the multiple off-line partition and the multiple off-line partition with adaptive load balancing, respectively. "LB" stands for the theoretical lower bound, which is one-third as there are three machines for the parallelization. The "MP" strategy, using off-line partitions, caused sub-optimal performance for some intervals, as a result of the significantly changed traffic pattern (probably due to the incident and the diversion of traffic). Before or after the incident, the "MP" still performed reasonably well. However, during the incident "MA" provided significant savings, which indicate that the use of the adaptive strategy is well justified.

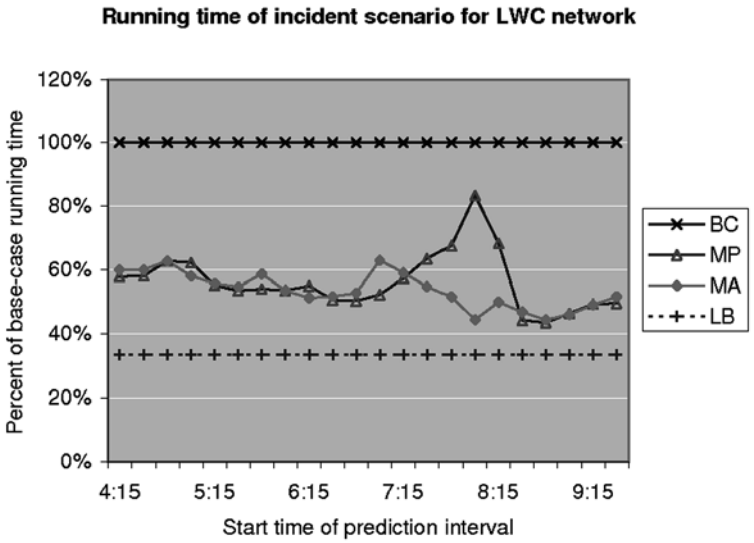


Fig. 10.23 Comparison of computing times for NY network under incident scenario

The case studies show that the online adaptive load balancing approach is more robust in terms of requiring less running time. Although using only off-line partitions may provide the best performance in normal situations, its "worst-case" performance (in terms of maximum computation time for a single horizon) was unacceptable. The adaptive approach showed reasonable speed-up ratios under various normal circumstances and better "worst-case" performance. Such a feature is important for online applications, where the system operates continuously and the simulation is expected to finish in a pre-specified amount of time regardless of the traffic conditions.

References

Antoniou C, Ben-Akiva M, Bierlaire M, Mishalani R (1997) Demand simulation for dynamic traffic assignment. Proceedings of the 8th IFAC symposium on transportation systems, Chania, Greece

Antoniou C, Ben-Akiva, Koutsopoulos HN (2007) Non-linear Kalman filtering algorithms for on-line calibration of dynamic traffic assignment models. IEEE Transactions on Intelligent Transportation Systems, vol 8. Issue 4, pp 661–670

Argonne National Laboratory (2008) MPICH2: high-performance and widely portable MPI. http://www.mcs.anl.gov/research/projects/mpich2/, Accessed 24 July 2008

Ashok K, Ben-Akiva M (1993) 'Dynamic O-D matrix estimation and prediction for real-time traffic management systems'. In: Daganzo C (ed) 'Transportation and traffic theory', Elsevier Science Publishing, Oxford, pp 465–484

Ashok K, Ben-Akiva M (2000) Alternative approaches for real-time estimation and prediction of time-dependent origin-destination flows. Transportation Science, vol 34, no.1

Balakrishna R (2006) Off-line calibration of dynamic traffic assignment models. PhD thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology

Balakrishna R, Koutsopoulos HN, Ben-Akiva M (2005) "Calibration and Validation of Dynamic Traffic Assignment Systems." Mahmassani HS (ed) 16th international symposium on transportation and traffic theory (ISTTT), Maryland, pp 407–426 (ISBN: 0-08-044680-9)

Balakrishna R, Ben-Akiva M, Koutsopoulos HN (2007) Off-line calibration of dynamic traffic assignment: simultaneous demand and supply estimation. Tran Res Record, No. 2003, 50–58

Balakrishna R, Wen Y, Ben-Akiva M, Antoniou C (2008) Simulation-based framework for transportation network management for emergencies. Transportation Res Record: J Trans Res Board. Number 2041, 80–88.

Ben-Akiva M, Bierlaire M, Bottom J, Koutsopoulos HN, Mishalani RG (1997) Development of a route guidance generation system for real-time application. Proceedings of the 8th IFAC symposium on transportation systems, Chania, Greece

Ben-Akiva M, Bierlaire M (1999) Discrete choice methods and their applications to short-term travel decisions. In: Hall R (ed) Handbook of transportation science, international series in operations research and management science, vol 23. Kluwer Academic, Boston

Ben-Akiva M, Bierlaire M, Koutsopoulos HN, Mishalani R (2002) 'Real-time simulation of traffic demand-supply interactions within DynaMIT'. In: Gendreau M, Marcotte P (eds) 'Transportation and network analysis: current trends. Miscellenea in honor of Michael Florian', Kluwer Academic Publishers, Boston/Dordrecht/London, pp 19–36

Ben-Akiva M, Bierlaire M (2003) Discrete choice models with applications to departure time and route choice. In: Hall R (ed) Handbook of transportation science, 2nd edn, Kluwer Academic, Boston

Ben-Akiva M, Koutsopoulos HN, Walker J (2001) DynaMIT-P. Dynamic assignment model system for transportation planning. Proceedings of the 2001 world conference in transportation research (WCTR), Seoul, Korea

Bottom J, Ben-Akiva M, Bierlaire M, Chabini I, Koutsopoulos HN, Yang Q (1999) Investigation of route guidance generation issues by simulation with DynaMIT. In: Ceder A (ed) Transportation and traffic theory, proceedings of the 14th international symposium on transportation and traffic theory, Pergamon, Oxford, pp 577–600

Cascetta E, Inaudi D, Marquis G (1993) Dynamic estimators of origin-destination matrices using traffic counts. Trans Sci 27(4):363–373

Cascetta E, Russo F, Viola FA, Vitetta A (2002) A model of route perception in urban road networks. Trans Res Part B: Methodological 36(7):577–592

Chui CK, Chen G (1999) Kalman filtering with real-time applications, Springer, New York

Greene WH (2000) Econometric analysis, 4th edn, Prentice-Hall Inc., Upper Saddle River, New Jersey

HCM (2000) Highway capacity manual. Transportation Research Board, Washington, DC

Kalman RE (1960) A new approach to linear filtering and prediction problems. J Basic Eng (ASME) 82D:35–45

Rathi V, Antoniou C, Wen Y, Ben-Akiva M, Cusack M (2008) Assessment of the impact of dynamic prediction-based route guidance using a simulation-based, closed-loop framework. Proceedings of the 87th annual meeting of the transportation research board, Washington, DC

Spall JC (1998) Implementation of the simultaneous perturbation algorithm for stochastic optimization. IEEE Trans on Aerospace Electronic Sys 34(3):817–823

Sundaram S (2002) Development of a dynamic traffic assignment system for short-term planning applications. Master's thesis, Massachusetts Institute of Technology.

UC Berkeley, Caltrans (2005) Freeway performance measurement system (PEMS) 5.4. http://pems.eecs.berkeley.edu/Public. Accessed 30th Apr 2009

Wen Y, Balakrishna R, Ben-Akiva M, Smith S (2007) “On-line deployment of dynamic traffic assignment: evaluations and lessons.” 11th world conference on transport research (WCTR), 24–28 June, Berkeley.

Wen Y (2009) Scalability of dynamic traffic assignment, PhD thesis, Massachusetts Institute of Technology

Xu S (2009) Development and test of dynamic congestion pricing model. Master thesis, Massachusetts Institute of Technology

Chapter 11
Traffic Simulation with METANET

Markos Papageorgiou, Ioannis Papamichail, Albert Messmer, and Yibing Wang

11.1 Introduction

Mathematical modeling is the imitation of the relevant aspects of a process (e.g., traffic flow) by use of appropriate mathematical equations and further logical relationships. When fed with sufficient initial and boundary conditions as well as control inputs and further exogenous variables, a *dynamic* model may produce the evolution of the process state over time. If the modeling equations are appropriately implemented in a computer, the resulting *simulator* may be used as a cost-effective and convenient tool for multiple uses, such as, in the case of traffic flow, planning of new or extended transportation infrastructures and comparison of alternatives; development and evaluation of various traffic control measures, strategies, and systems; short-term prediction and surveillance of traffic state in complex networks; evaluation of the impact of capacity-reducing events (e.g., road works, incidents), or increased demand, etc.

Two basic modeling approaches have been pursued in the area of traffic flow: microscopic modeling, which describes the longitudinal (car-following) and lateral (lane-changing) movement of individual vehicles, and macroscopic modeling, which addresses traffic as a particular fluid with aggregate variables (density, mean

M. Papageorgiou (✉)
Dynamic Systems and Simulation Laboratory, Technical University of Crete, Chania 73100, Greece
e-mail: markos@dssl.tuc.gr

I. Papamichail
Dynamic Systems and Simulation Laboratory, Technical University of Crete, Chania 73100, Greece
e-mail: ipapa@dssl.tuc.gr

A. Messmer
Independent Engineer Groebenseeweg 2, D-82402 Seeshaupt, Germany
e-mail: albert.messmer@t-online.de

Y. Wang
Department of Civil Engineering, Monash University, Victoria 3800, Australia
e-mail: yibing.wang@monash.edu

J. Barceló (ed.), *Fundamentals of Traffic Simulation*,
International Series in Operations Research & Management Science 145,
DOI 10.1007/978-1-4419-6142-6_11,

speed, flow). Both approaches may be contrasted with respect to a number of aspects:

- *Simulation*: The traffic flow simulation market is dominated by the microscopic approach, possibly due to its direct similarity with the perceived real process as compared to the abstract, mathematically more challenging macroscopic description.
- *Computational effort* is far lower in macroscopic approaches.
- Beyond simulation, macroscopic models of whole traffic networks may be expressed in an *analytic form* that opens the way for the use of a multitude of powerful available surveillance and control methods such as Kalman Filter, Automatic Control methods, and gradient-based optimization.
- *Accuracy* depends on the employed validation procedures that are easier to apply in the low-resolution macroscopic models.
- Specific considerations (e.g., impact of various messages to the driver, mixed automated/non-automated traffic flow, and impact of infrastructure layout changes) may be more easily and directly incorporated in microscopic models.

Since the legendary paper by Lighthill and Whitham (1955), a number of various dynamic macroscopic models, mostly in the form of partial differential equations (PDE), have been proposed. The interested reader is referred to (Hoogendoorn and Bovy 2001) for an overview. As the conservation equation is the only exact relationship in traffic flow modeling, it is included in all approaches. In addition, first-order models involve a static speed–density relationship while second-order models address the mean speed dynamics with potentially more realism. Although the technical literature on macroscopic traffic flow modeling is vast and increasing in an accelerated pace, rigorous model validation exercises using real traffic data are surprisingly sparse. Given the largely empirical character of the proposed models, the lack of validation efforts is a shortcoming that cannot be sufficiently emphasized.

Another issue connected to macroscopic models is the space–time discretization of the related PDE in order to enable their numerical solution in digital computers. In many cases, sophisticated numerical schemes are employed for a reliable and accurate numerical approximation of the PDE. These approaches, however, typically result in complex computational schemes that require a high computational effort and, moreover, they may not lead to analytical discretized models; in other words, these approaches employ a significant effort to approximate the PDE that are all but exact. An alternative, more practicable approach is to discretize the original empirical PDE by use of simple (e.g., Euler) schemes leading to analytical state-space models which may then be readily validated; implemented in a computer with low computational effort; and used as a valuable basis for the analytical derivation of various surveillance and control tools. The main disadvantage of these approaches is due to the fact that any theoretical investigations and results obtained for the PDE are not directly transferable to the discretized model.

This chapter presents METANET (Messmer and Papageorgiou 1990), a program for motorway network simulation based on a purely macroscopic modeling approach. This leads to relatively low computational effort, which is independent of the load (number of vehicles) in the simulated network and allows also for a

real-time use of the model. The overall modeling approach allows for simulation of all kinds of traffic conditions (free, dense, and congested) and of capacity-reducing events (incidents) with prescribed characteristics (location, intensity, and duration).

METANET may be applied to existing or hypothetical, multi-origin, multi-destination, multi-route motorway networks with arbitrary topology and geometric characteristics including bifurcations, junctions, on-ramps, and off-ramps. By use of a special modeling option (store-and-forward links), METANET provides also the possibility to consider non-motorway links in a simplified way. An extension of METANET, which offers the possibility to also address signal-controlled urban road networks based on a realistic macroscopic modeling approach for urban links and junctions, is METACOR (Elloumi et al., 1994; Diakaki and Papageorgiou 1996), a macroscopic modeling tool for urban corridors.

METANET considers the application of traffic control measures, such as route guidance, ramp metering, motorway-to-motorway control, variable speed limits, shoulder lane opening or closing, and signal-based mainstream traffic control at arbitrary network locations. Several options are offered for describing or prescribing the average route choice behavior of driver groups with particular destinations. Route guidance and dynamic traffic assignment considerations in METANET are based on the notion of splitting rates at bifurcation nodes rather than on path assignment. Among other advantages, this approach enables consideration of route guidance or traffic assignment for a part of the network (rather than the whole network) if so desired by the user.

Simulation results are delivered in terms of macroscopic traffic variables such as traffic density, traffic volume, and mean speed at all network locations as well as in terms of travel times on selected routes. This is done for a configurable output time interval that is chosen usually significantly longer (typically 1 min) than the simulation time step (typically 5–20 s). Global evaluation indexes such as total travel time, total traveled distance, total fuel consumption, total waiting time at network origins, and total disbenefit of routed drivers are also calculated. For displaying traffic data generated or used by METANET in a transparent form, a specific graphical output program called METAGRAF is available. Visualization of results is provided both by time trajectories of selected variables and by graphical representation of the whole network.

The very first version of METANET was developed in 1989, and the tool has been adopted since that time by more than 100 user institutions from academia, administrations, and industry. A full documentation of the tool comprising a full description of the modeling approach and detailed instructions for the software use is available for users.

11.2 Model Building Principles in METANET

The motorway network is represented by a directed graph whereby the links of the graph represent motorway stretches with uniform characteristics, i.e., no on-/off-ramps and no major changes in geometry. The nodes of the graph are placed

at locations where a major change in road geometry occurs, as well as at bifurcations, junctions, on-ramps, and off-ramps. The macroscopic description of traffic flow implies the definition of adequate variables expressing the aggregate behavior of traffic at certain times and locations while the time and space arguments are discretized.

METANET considers five types of links (normal motorway links, origin links, store-and-forward links, destination links, and dummy links) and each one is used for different reasons and thus treated in a distinct way. The two directions of a motorway stretch are modeled as separate links with opposite directions. Inside each link we suppose homogeneous geometric characteristics such as number of lanes, grade, and curvature. An inhomogeneous motorway stretch may be represented by two or more consecutive links separated by nodes at the locations where the change of geometry occurs. At the bounds of the network, origin or destination links are added where traffic enters or leaves, respectively, the simulated network part. METANET offers the possibility to model incidents at user-specific locations with user-specific occurrence time and duration via appropriate modification of the ordinary model equations.

Essentially the simulation is fed with demands at its boundaries (inflows) plus origin–destination information (if necessary). These data together with other values which appear at the network bounds (speeds at main inflows and traffic densities at main outflows) are called *boundary data* (or input traffic data) in the following. The origin of the data may be from measurements (if real traffic situations are reconstructed), or the data may be hypothetical (if certain types of traffic situations are studied). The files in which boundary data are provided may contain also data from inside the network which are not considered by METANET but which can be used by METAGRAF for comparison purposes.

Each origin–destination couple may be connected by one or more routes. Based on the network topology, METANET finds automatically all possible loop-free routes. The route choice behavior inside the network is described by use of *splitting rates* which express the portion of drivers deciding at a bifurcation node to use a certain alternative output link toward their destination. Splitting rates can be looked upon as turning rates (the ratio of the traffic volumes in each output link of a node) *by destination*.

For simulation of control measures via variable direction indication and also for dynamic traffic assignment studies, the destination-oriented way of modeling, as outlined above and described in more detail later, is necessary. However, for many applications where route guidance is not involved, the user may not be interested in the composition of traffic in terms of destinations. In this case no origin–destination information is necessary, and moreover no splitting rates, but simply turning rates, are needed at the network nodes.

In case of multiple routes connecting any two nodes in the network, METANET offers the possibility to model the routing behavior of drivers based on dynamic equilibrium assignment which is approximated by use of feedback concepts (Section 11.4). Finally, METANET may consider the impact of various control measures (installed at user-specific locations) including the following:

- Variable direction signs for route guidance;
- Queue information via variable message signs (VMS) that may alter driver routing;
- Lane closures;
- Shoulder lane opening;
- Variable speed limits (VSL);
- Ramp metering;
- Motorway-to-motorway control;
- Traffic lights on the mainstream or at the off-ramps.

Clearly, in order to operate these control measures as in real time, corresponding control strategies are needed. METANET offers (by default) some selected feedback strategies (e.g., the ALINEA local ramp metering strategy by Papageorgiou et al. (1997)) for some of these control measures for the convenience of the user. However, the user may introduce or program any other control strategy for any of these control measures in a special METANET module reserved to this end.

In order to assess and compare the efficiency of various control measures and/or control strategies, some global *performance criteria* are calculated over the simulated time horizon:

- total travel time;
- total waiting time (in the queues of the origin links and of the store-and-forward links);
- total time spent in the network;
- total distance traveled;
- total amount of fuel consumed; and
- total disbenefit of routed drivers.

The METANET program is written in C. With slight modifications, it can be compiled on any machine equipped with an ANSI-C compiler. Compilation is tested and according make-files are available for UNIX (system V) and for MS-Windows using the Visual C++ compiler. For PC (under Windows) an already compiled exe-file (running in a console window) is delivered with METANET.

METANET uses dynamic memory allocation for the bigger data structures, i.e., at run time, after reading the network configuration, the structures containing network information and traffic data are allocated according to the actually required size. This helps for better utilization of available memory, thus enabling the simulation of very large-scale networks.

A simulation run starts with reading the file for simulation control and the file which describes the network to be simulated. In order to give the user the opportunity to check the input, all data together with additional information generated by the program are written to a check file. Numerous input data checks are performed already by the program. If inconsistent input data are found or if information is missing, an error message is displayed and the program terminates.

After initialization, the main simulation loop starts. For each time step the simulation is fed with new boundary traffic data, control inputs, and splitting rates which may be interpolated from data provided in the corresponding traffic data files. Interpolation is necessary if the time raster of provided boundary data is coarser than the simulation time step of, e.g., 10 s. In a user-specified time raster, the currently calculated traffic variables are then written to an output file. The user may specify a subset of all available traffic variables which should be written to the output.

The simulation loop is terminated after a predefined number of steps or if an end of file is encountered in one of the traffic data input files. At the end of the simulation run, the shortest travel times along the different directions of user-selected bifurcation nodes over the simulation horizon may be calculated. Some final information (e.g., performance criteria) is written to the output file and a balance (by destination) over all vehicles that entered and left the network is made (considering the vehicles still being in the network). An error message is issued if balances deviate significantly from zero.

As already mentioned in Section 11.1, for displaying the traffic data generated or used by METANET in a transparent form, a graphical output program called METAGRAF is available. METAGRAF offers four main options:

1. Plot of network topology;
2. Network traffic overview at given time instances;
3. Line plots of traffic flows, speeds, densities, queue lengths, and composition rates over time for selected links;
4. Line plots of splitting rates, node inflows, travel times, and disbenefit values over time for selected (bifurcation-destination)-couples.

METAGRAF is a Windows9x application. User input is done via self-explaining menus.

11.3 Core Traffic Flow Models

As explained in the previous section, the motorway network is represented by a directed graph consisting of links and nodes. The traffic flow models used for each of the five types of links (normal motorway links, origin links, store-and-forward links, destination links, and dummy links) and the nodes are described in what follows. METANET has two distinct modes of operation. When traffic assignment aspects are not considered, then it may be operated in the non-destination-oriented mode (this section). When traffic assignment is an issue, it must be operated in the destination-oriented mode (see Section 11.4).

11.3.1 Links

Normal Motorway Links. A second-order macroscopic discretized traffic flow model is used for the description of traffic flow on a normal motorway link. This model is suitable for free-flow, critical, and congested traffic conditions. The macroscopic description of traffic flow implies the definition of adequate variables expressing the aggregate behavior of traffic at certain times and locations. The time and space arguments are discretized. The discrete time step is denoted by T (typically $T = 10$s). A motorway link m is divided into N_m segments of equal length L_m (typically $L_m = 500$ m) (Fig. 11.1), such that the numerical stability condition $L_m \geq Tv_{f,m}$ holds, where $v_{f,m}$ is the free speed on link m. Each segment i of link m at discrete time $t = kT$, $k = 0, 1, ..., K$, where K is the time horizon, is macroscopically characterized via the following variables:

- *Traffic density* $\rho_{m,i}(k)$ (veh/km/lane) is the number of vehicles in segment i of link m at time $t = kT$ divided by L_m and by the number of lanes λ_m.
- *Mean speed* $v_{m,i}(k)$ (km/h) is the mean speed of the vehicles included in segment i of link m at time $t = kT$.
- *Traffic volume* or *flow* $q_{m,i}(k)$ (veh/h) is the number of vehicles leaving segment i of link m during the time period $[kT, (k+1)T)$, divided by T.

In the non-destination-oriented model, the previously defined traffic variables are calculated for each segment i of link m at each time step k by the following equations:

$$\rho_{m,i}(k+1) = \rho_{m,i}(k) + \frac{T}{L_m \lambda_m}\left[q_{m,i-1}(k) - q_{m,i}(k)\right] \tag{11.1}$$

$$q_{m,i}(k) = \rho_{m,i}(k)\, v_{m,i}(k)\, \lambda_m \tag{11.2}$$

$$\begin{aligned} v_{m,i}(k+1) = v_{m,i}(k) &+ \frac{T}{\tau}\left\{V\left[\rho_{m,i}(k)\right] - v_{m,i}(k)\right\} \\ &+ \frac{T}{L_m}\left[v_{m,i-1}(k) - v_{m,i}(k)\right] v_{m,i}(k) - \frac{\nu T}{\tau L_m}\frac{\rho_{m,i+1}(k) - \rho_{m,i}(k)}{\rho_{m,i}(k) + \kappa} \end{aligned} \tag{11.3}$$

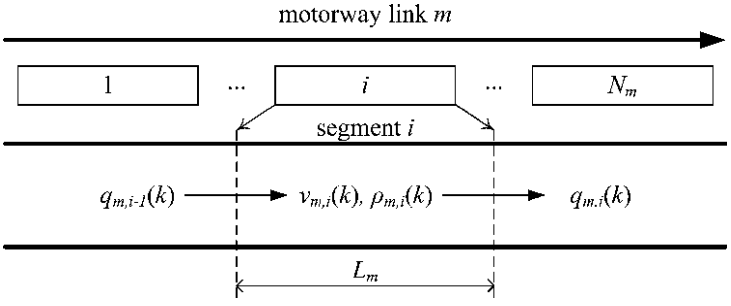


Fig. 11.1 Discretized motorway link

$$V\left[\rho_{m,i}(k)\right] = v_{f,m} \exp\left[-\frac{1}{\alpha_m}\left(\frac{\rho_{m,i}(k)}{\rho_{cr,m}}\right)^{\alpha_m}\right] \tag{11.4}$$

where $\rho_{cr,m}$ denotes the critical density per lane of link m and α_m is a parameter of the fundamental diagram (eq. (11.4)) of link m expressing a monotonically decreasing nonlinear relationship between the mean speed and the traffic density. Furthermore, the time constant τ, the anticipation constant ν, and κ are further parameters that are equal for all the network links. The parameter values can be determined via a validation procedure described in Section 11.5.

This second-order model was essentially proposed by Payne (1971). Equation (11.1) expresses the vehicle conservation principle, while eq. (11.2) is the flow equation which results directly from the non-discretized definition of the traffic variables. Equation (11.3) is an empirical dynamic speed equation which describes the dynamic evolution of the mean speed of each segment as an independent variable. Equation (11.4) is also an empirical static relationship between speed and traffic density. Two additional terms may be included in eq. (11.3) for the modeling of lane drops and merging phenomena at on-ramps as suggested by Papageorgiou et al. (1990), which include two constant model parameters δ and ϕ. Additionally, the mean speed resulting from eq. (11.3) is limited from below by the minimum speed $v_{\min}$ in order to avoid unrealistically low flows during congestion.

The described link model was extended by Papamichail et al. (2008) to incorporate the impact of displayed VSL values on the traffic flow behavior under the assumption that a single VSL value (if any) is displayed in each link. Particular VSL values are reflected in the link-specific *VSL rates* $b_m(k)$ that prevail, by definition, during $[kT, (k+1)T)$. The VSL rates are control variables with an admissible value range $b_m(k) \in \left[b_{\min,m}, 1\right]$ where $b_{\min} \in (0, 1)$ is a lower admissible bound for VSL rates (see here below for a physical interpretation of b_m).

The VSL rates were included into the link model expressed by eqs. (11.1), (11.2), (11.3), and (11.4) by rendering the static speed–density relationship of eq. (11.4) b_m dependent, that is, by actually rendering the three parameters included in eq. (11.4) b_m dependent with the use of the following affine functions:

$$v_{f,m}\left[b_m(k)\right] = v^*_{f,m} b_m(k) \tag{11.5}$$

$$\rho_{cr,m}\left[b_m(k)\right] = \rho^*_{cr,m}\left\{1 + A_m\left[1 - b_m(k)\right]\right\} \tag{11.6}$$

$$\alpha_m\left[b_m(k)\right] = \alpha^*_m\left[E_m - (E_m - 1)\, b_m(k)\right] \tag{11.7}$$

where $v^*_{f,m}$, $\rho^*_{cr,m}$, α^*_m denote the specific non-VSL values for these parameters, while A_m, E_m are constant parameters to be estimated based on real data.

As eq. (11.5) reveals, b_m is equal to the VSL-induced $v_{f,m}$ divided by the non-VSL $v^*_{f,m}$, or, approximately, equal to the displayed VSL divided by the legal speed limit without VSL. Thus, if $b_m(k) = 1$, no VSL is applied, else $b_m(k) < 1$, and, in fact, for $b_m(k) = 1$, all parameters are seen in eqs. (11.5), (11.6), and (11.7) to

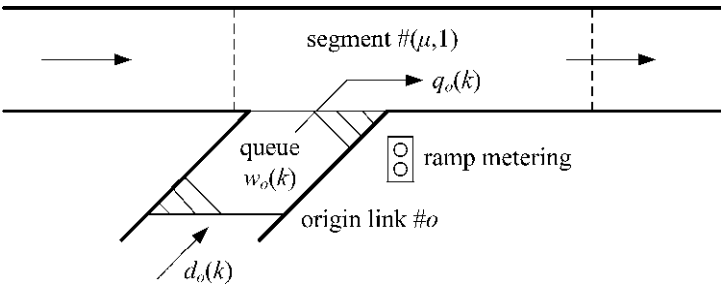


Fig. 11.2 The origin link queue model

attain their respective non-VSL values. Equations (11.6) and (11.7) suggest that, for $A_m > 0$ and $E_m > 1$, $\rho_{cr,m}$ and α_m are linear increasing functions for decreasing b_m, starting with their usual non-VSL values for $b_m(k) = 1$.

Origin Links. For origin links, i.e., links that receive traffic demand d_o and forward it into the motorway network, a simple queue model is used (Fig. 11.2). The outflow q_o of an origin link o depends on the traffic conditions of the corresponding mainstream segment $(\mu, 1)$ and the existence of ramp metering control measures. If ramp metering is applied, then the outflow $q_o(k)$ that leaves origin o during period k is a portion $r_o(k)$ of the maximum outflow $\hat{q}_o(k)$ that would leave in absence of ramp metering. If $r_o(k) = 1$, no ramp metering is applied, else $r_o(k) < 1$. Thus, $r_o(k) \in \left[r_{\min,o}, 1\right]$ is the metering rate for the origin link o, i.e., a control variable, where $r_{\min,o}$ is a minimum admissible value. The queuing model is described by the following conservation equation:

$$w_o(k+1) = w_o(k) + T\left[d_o(k) - q_o(k)\right] \tag{11.8}$$

where $w_o(k)$ (veh) is the queue length in origin o at time kT and $d_o(k)$ (veh/h) is the demand flow at o. The outflow q_o is determined as follows:

$$q_o(k) = r_o(k)\,\hat{q}_o(k) \tag{11.9}$$

with

$$\hat{q}_o(k) = \min\left\{\hat{q}_{o,1}(k), \hat{q}_{o,2}(k)\right\} \tag{11.10}$$

and

$$\hat{q}_{o,1}(k) = d_o(k) + w_o(k)/T \tag{11.11}$$

$$\hat{q}_{o,2}(k) = Q_o \min\left\{1, \frac{\rho_{\max} - \rho_{\mu,1}(k)}{\rho_{\max} - \rho_{cr,\mu}}\right\} \tag{11.12}$$

where Q_o (veh/h) is the on-ramp's flow capacity, i.e., the on-ramp's maximum possible outflow under free-flow traffic conditions in the mainstream, and $\rho_{\max}$ is the

maximum density in the network. Thus, the maximum outflow $\hat{q}_o(k)$ is determined by the current origin demand if $\hat{q}_{o,1}(k) < \hat{q}_{o,2}(k)$, or by the geometrical capacity Q_o if the mainstream density is undercritical, i.e., $\rho_{\mu,1}(k) < \rho_{cr,\mu}$, or by the reduced capacity due to congestion of the mainstream if $\rho_{\mu,1}(k) > \rho_{cr,\mu}$.

Store-and-Forward Links. For a number of reasons, among which is the modeling of motorway-to-motorway control measures and the simplified consideration of non-motorway routes with limited capacity, a similar simple queue model as above may also be used for some internal network links, called store-and-forward links. Traffic entering a store-and-forward link is added, after a constant travel time to the link queue and eventually forwarded to the next downstream node. Queue spillback is modeled to have an impact on the traffic flow of upstream links.

Destination Links. Traffic conditions in destination links are influenced by the downstream traffic conditions which may be provided as boundary conditions for the whole time horizon. Options for boundary conditions at destination links include free outflow (in which case there is no influence of the modeled network traffic from downstream traffic conditions) or pre-specified maximum possible outflow (in which case a queue may build if the arriving flow exceeds the maximum value) or boundary traffic density (which influences the upstream traffic conditions accordingly).

Dummy Links. Dummy links are auxiliary links of zero length. They do not affect traffic dynamics but are used to decompose complex nodes or to represent very short motorway connections.

11.3.2 Nodes

Motorway bifurcations and junctions (including on-ramps and off-ramps) are represented by nodes. Traffic enters a node n through a number of input links and is distributed to the output links according to the following equations:

$$Q_n(k) = \sum_{\mu \in I_n} q_{\mu,N_\mu}(k) \tag{11.13}$$

$$q_{m,0}(k) = \beta_n^m(k)\, Q_n(k)\ \forall m \in O_n \tag{11.14}$$

where I_n is the set of links entering node n, O_n is the set of links leaving n, $Q_n(k)$ is the total traffic volume entering n at period k, $q_{m,0}(k)$ is the traffic volume that leaves n via outlink m, and $\beta_n^m(k)$ is the portion of $Q_n(k)$ that leaves n through link m. Thus, $\beta_n^m(k),\ m \in O_n$, are the turning rates of node n.

If a node n has more than one leaving links, then the upstream influence of density has to be taken into account in the last segment of the incoming link (see eq. (11.3) for $i = N_m$). This is provided via

$$\rho_{m,N_m+1}(k) = \sum_{\mu \in O_n} \rho_{\mu,1}^2(k) \Big/ \sum_{\mu \in O_n} \rho_{\mu,1}(k) \tag{11.15}$$

where ρ_{m,N_m+1} is the virtual density downstream of the entering link m to be used in eq. (11.3) for $i = N_m$ and $\rho_{\mu,1}(k)$ is the density of the first segment of the leaving link μ. The quadratic average is used to account for the fact that traffic flow of one congested link may spill back into the entering link even if there is free flow in the other leaving link.

If a node n has more than one entering links, then the downstream influence of speed has to be taken into account according to eq. (11.3) for $i = 1$. The mean speed value is calculated from

$$v_{m,0} = \sum_{\mu \in I_n} v_{\mu,N_\mu}(k)\, q_{\mu,N_\mu}(k) \Big/ \sum_{\mu \in I_n} q_{\mu,N_\mu}(k) \tag{11.16}$$

where $v_{m,0}$ is the virtual speed upstream of the leaving link m that is needed in eq. (11.3) for $i = 1$.

11.3.3 Model Summary

Combining the equations developed in the previous sections, a nonlinear analytic discrete-time macroscopic model of the following state-space form:

$$\mathbf{x}(k+1) = \mathbf{f}\big[\mathbf{x}(k), \mathbf{u}(k), \mathbf{d}(k), \mathbf{p}\big], \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{11.17}$$

can be obtained for the entire motorway network, where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the control vector, $\mathbf{d}$ is the disturbance vector, and $\mathbf{p}$ is the parameter vector. This model can be used to simulate the motorway network traffic as shown in Fig. 11.3 and to test various control strategies.

In the non-destination-oriented mode, the state vector consists of the densities $\rho_{m,i}$ and the mean speeds $v_{m,i}$ of every segment i of every link m and the queues w_o of every origin and store-and-forward link o. The control vector consists of the VSL rates b_m of every link m where VSL is applied and the ramp metering rates r_o of every origin and store-and-forward link o that is metered; the modeling of further

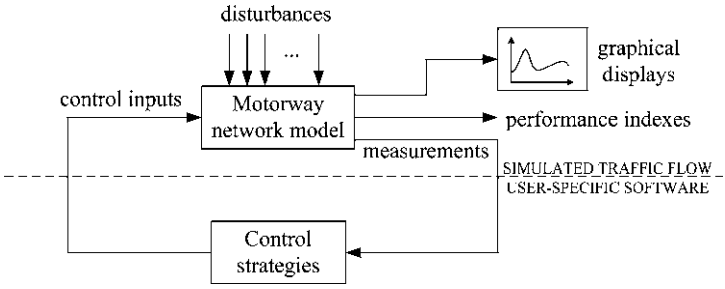


Fig. 11.3 The control loop

control measures mentioned in Section 11.2 is not detailed here for space limitation. The disturbance vector consists of the demand d_o at every origin o, the turning rates β_n^m at every bifurcation node n, and all boundary variables. Finally, the parameter vector consists of the free speed $v_{f,m}^*$, the critical density $\rho_{cr,m}^*$ and α_m^* for every link m, and the parameters τ, ν, κ, α, and δ that are common for all the network links. In addition the parameters A_m, E_m must be specified if variable speed limits are included.

This state-space formulation is particularly valuable since it allows for the use of well-known methods for the estimation and prediction of the state and parameter vectors as well as for the design of feedback and dynamic optimal control strategies for the motorway network traffic (see Section 11.6).

11.3.4 User-Programmable Control Strategies

In presence of control measures, the introduced control variable (as well as route guidance-related variables of Section 11.4) must be specified, as they are external to the model. Control variables may be pre-specified (constant or time variant) in corresponding input files, but, in most cases, the user may be interested in testing or comparing specific related control strategies. As an option, METANET incorporates some specific control strategies (e.g., the ALINEA local ramp metering algorithm) for the user convenience. Otherwise, a special program module is reserved for user-programmed traffic-responsive control strategies. At each simulation time step this module may use any available traffic variable in order to calculate the control actions "in real time." Traffic variables are global and hence the control strategy can use any simulated traffic variable as "real-time measurement" (Fig. 11.3).

11.4 Dynamic Traffic Assignment

Motorway networks may include a large number of origins and destinations with multiple paths connecting each origin–destination pair. During rush hours, the travel time on many routes changes substantially due to traffic congestion, and alternative routes may become competitive. Drivers who are familiar with the traffic conditions in a network (e.g., commuters) optimize their individual routes based on their past experience, thus leading to user-equilibrium conditions, first proposed by Wardrop (1952). These conditions suggest that the travel times of any same-destination drivers leaving any bifurcation node at the same time interval but using different directions will be equal. The task of specifying dynamic equilibrium conditions in a traffic network via appropriate routing of sub-flows is known as dynamic traffic assignment and is supposed to model the real routing behavior of network-experienced drivers (e.g., commuters) in an aggregated way. On the other hand, daily varying demands, changing environmental conditions, exceptional events, and incidents may change the traffic conditions in an unpredictable way. This may lead

to an underutilization of the overall network′s capacity, whereby some links are heavily congested while capacity reserves are available on alternative routes. Route guidance and driver information systems may be employed to improve the network efficiency via direct or indirect recommendation of alternative routes. Thus, dynamic traffic assignment and route guidance address essentially the same problem (routing) albeit with different application conditions (off-line modeling versus real-time control).

When traffic assignment or route guidance is an issue, the second-order macroscopic model must be operated in the destination-oriented mode in which the following additional variables are introduced:

- *Partial density* $\rho_{m,i,j}(k)$ is the density of vehicles in segment i of link m at time $t = kT$ destined to destination $j \in J_m$, where J_m is the set of destinations reachable via link m.
- Composition rate $\gamma_{m,i,j}(k)$, $0 \leq \gamma_{m,i,j}(k) \leq 1$, is the portion of traffic volume $q_{m,i}(k)$ or traffic density $\rho_{m,i}(k)$ which is destined to destination $j \in J_m$.

The notion of turning rates β_n^m is generalized to the notion of splitting rates. Let $Q_{n,j}(k)$ be the total traffic volume entering motorway node n at period k, which is destined to j. Then the splitting rate $\beta_{n,j}^m(k)$ is the portion of the traffic volume $Q_{n,j}(k)$ which leaves node n at period k through link $m \in O_n$, hence $0 \leq \beta_{n,j}^m(k) \leq 1$. Then, for any network node

$$Q_{n,j}(k) = \sum_{\mu \in I_n} q_{\mu,N_\mu}(k)\, \gamma_{\mu,N_\mu,j}(k) \quad \forall (n,j) \tag{11.18}$$

$$q_{m,0}(k) = \sum_{j \in J_m} Q_{n,j}(k)\, \beta_{n,j}^m(k) \quad \forall m \in O_n \tag{11.19}$$

$$\gamma_{m,0,j}(k) = \beta_{n,j}^m(k)\, Q_{n,j}(k) / q_{m,0}(k) \quad \forall m \in O_n,\ \forall j \in J_m \tag{11.20}$$

Note that $\sum_{m \in O_n} \beta_{n,j}^m(k) = 1$ holds, which reduces the number of independent splitting rates at each bifurcation node by one.

The partial densities in every link are calculated from conservation considerations

$$\rho_{m,i,j}(k+1) = \rho_{m,i,j}(k) + \frac{T}{L_m \lambda_m}\left[\gamma_{m,i-1,j}(k)\, q_{m,i-1}(k) - \gamma_{m,i,j}(k)\, q_{m,i}(k)\right] \quad \forall j \in J_m \tag{11.21}$$

while $\gamma_{m,i,j}(k) = \rho_{m,i,j}(k) / \rho_{m,i}(k)$. Similarly, for the queues of origin and store and forward links the notion of partial queues $w_{o,j}$ is introduced. Partial queues evolve according to the relationship

$$w_{o,j}(k+1) = w_{o,j}(k) + T\left[\theta_{o,j}(k)\, d_o(k) - \gamma_{o,j}(k)\, q_o(k)\right] \tag{11.22}$$

where $\theta_{o,j}(k)$ is the portion of $d_o(k)$ destined to j during period k, $\gamma_{o,j} = w_{o,j}(k)/w_o(k)$, and $w_{o,j}(k)$ is the number of vehicles in the queue of origin link o with destination j.

In the case where route guidance takes place at a node n with respect to destination j (using variable message signs or other means), a direction is recommended to the drivers toward this destination. This recommendation may affect the drivers' behavior, depending on their compliance rate. Since the routing message refers to particular destinations, the influence on the route choice is projected to the corresponding splitting rates of the node. At bifurcation node n for destination j, the following splitting rates are defined: $\beta^m_{N,n,j}(k)$ is the nominal spitting of drivers in absence of any guidance; $\beta^m_{G,n,j}(k)$ is the spitting order by the system, i.e., a control variable; $\beta^m_{n,j}(k)$ is the real spitting according to drivers compliance. The relation between these three quantities is modeled by the following equation:

$$\beta^m_{n,j} = (1-\varepsilon)\,\beta^m_{N,n,j} + \varepsilon\,\beta^m_{G,n,j} \tag{11.23}$$

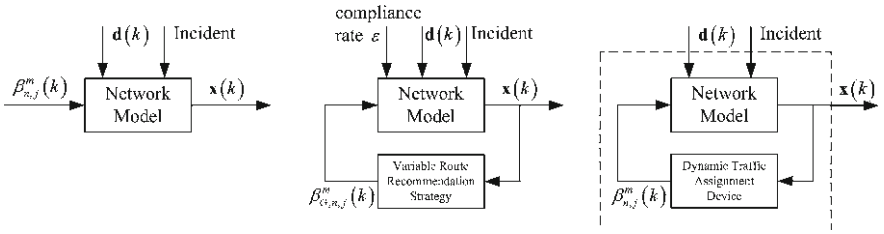
where ε is the compliance rate to the guidance instructions ($0 \le \varepsilon \le 1$).

The described approach uses splitting rates (i.e., turning rates by destination) for traffic assignment or route guidance purposes, while the use of paths connecting each origin–destination pair in the network is a more popular approach. The advantages of splitting rates over paths include the following:

- The number of splitting rates in large networks may be orders-of-magnitude smaller than the number of paths with corresponding implications for simplicity and computation times.
- A splitting rate $\beta^m_{n,j}(k)$ has a direct physical interpretation: It corresponds to the indications of a VMS located at network node n and guiding drivers bound for destination j. Moreover, splitting rates may be changed while the simulator is running so as to reflect the decisions of a corresponding route guidance strategy.

Clearly, path-based flows may be readily translated to corresponding splitting rate values while the opposite is not always possible. In fact, awkward values of splitting rates may lead to cyclic vehicle routes; however, proper dynamic traffic assignment algorithms for user optimum or system optimum are not expected to lead to such unrealistic phenomena that would increase the resulting travel cost.

Combining the equations developed in the previous sections, a nonlinear discrete-time macroscopic model of the state-space form of eq. (11.17) can again be obtained for the entire motorway network. In the destination-oriented mode the state vector of eq. (11.17) consists of the partial densities $\rho_{m,i,j}$ of every segment i and reachable destination j from link m, the mean speeds $v_{m,i}$ of every segment i of every link m, and the partial queues $w_{o,j}$ of every origin and store-and forward link o. The control vector consists of the VSL rates b_m of every link m where VSL is applied, the ramp metering rates r_o of every origin and store-and forward link o that is metered, and the splitting rates $\beta^m_{G,n,j}$ at every bifurcation node n where route guidance with respect to destination j takes place. The disturbance vector consists of the demand d_o at



(a) Simulation with given $\beta(k)$ (b) Control Strategy Testing (c) Simulation including Dynamic Traffic Assignment

Fig. 11.4 User options for splitting rate specification

every origin o, the composition rates $\theta_{o,j}$ (OD matrix), the spiting rates $\beta^m_{n,j}$ at every bifurcation node n where no route guidance is applied, the nominal splitting rates $\beta^m_{N,n,j}$ for every bifurcation node n where route guidance with respect to destination j takes place, and the drivers' compliance rate.

The described destination-oriented model version enables the consideration of dynamic traffic assignment or route guidance if the introduced splitting rates are appropriately specified. Figure 11.4 illustrates three available user options:

- The user may enter splitting rate values (time-variant or constant) via a corresponding input file (Fig. 11.4a). This option may be used for testing of specific driver routing scenarios. Note that, for networks without multiple routes (e.g., a motorway axis with on-ramps and off-ramps), no splitting rates are needed, but the destination-oriented model version may still be used for a more accurate OD-dependent traffic flow simulation.
- The user may program an own real-time route guidance strategy applying at prespecified (bifurcation-destination) couples (so-called (n,j)-couples) according to Fig. 11.3, i.e., delivering ordered splitting rates that are used in eq. (11.23) along with a fixed compliance rate (Fig. 11.4b). In this case splitting rates must also be entered (via input file) for all (n,j)-couples that are not addressed by the route guidance strategy.
- If the user is interested in dynamic equilibrium conditions, METANET can calculate automatically the corresponding splitting rates for all (n,j)-couples using appropriate feedback algorithms proposed by Papageorgiou (1990) and Messmer and Papageorgiou (1994) which are based on instantaneous (reactive) travel times. Note that this leads to an approximate dynamic equilibrium since no iterations are involved, see (Wang et al., 2001), for more details. This option is illustrated in Fig. 11.4c.

11.5 Calibration and Validation

The model validation procedure aims at enabling the macroscopic model of the motorway network to represent the real traffic conditions with sufficient accuracy, mainly via appropriate specification of the parameters included in the model

equations. The estimation of the unknown parameters included in eq. (11.17) is not a trivial task, since the system equations are highly nonlinear in both the parameters and the state variables. The most common approach is the minimization of the discrepancy between the model calculations and the real process in the sense of a quadratic output error functional (Cremer and Papageorgiou 1981; Papageorgiou et al., 1989, 1990; Kotsialos et al., 2002a; Papamichail et al., 2007). Let $\mathbf{y}$ be the measurable output vector (typically consisting of flows and mean speeds at various locations) of the nonlinear system (11.17) given by

$$\mathbf{y}(k) = \mathbf{g}\left[\mathbf{x}(k), \mathbf{p}\right]. \tag{11.24}$$

Then, the parameter estimation problem can be formulated as a least-squares output error problem as follows: Given the disturbance and the control vectors as well as the measured process output $\mathbf{y}^m(k)$ for some $k \in \mathrm{M} \subseteq \{1, 2, ..., K\}$, and the initial state $\mathbf{x}_0$, find the set of parameters $\mathbf{p}$ minimizing the cost functional

$$J(\mathbf{p}) = \sum_{k \in \mathrm{M}} \left\| \mathbf{y}(k) - \mathbf{y}^m(k) \right\|_{\mathbf{Q}}^2 \tag{11.25}$$

subject to eqs. (11.17) and (11.24), where $\mathbf{Q}$ is a positive definite, diagonal matrix (Fig. 11.5). M may be a subset of the simulated points, as the simulation step is set to a value, e.g., $T = 10$ s, that is less than the utilized measurement interval, e.g., 60 s. The model parameters are selected from a closed admissible region of the parameter space which may be defined on the basis of physical considerations. The determination of the optimal parameter set must be performed by means of a nonlinear programming routine. Papageorgiou et al. (1990) demonstrated that the model is most sensitive with respect to the values of the parameters used in the static speed–density eq. (11.4).

As an example, this model validation procedure has been applied by Kotsialos et al. (2002a) to the Amsterdam motorway network shown in Fig. 11.6. Each motorway was modeled in both directions as shown in Fig. 11.7 (produced by METAGRAF). The total length of the network is 143 km and its main part is the A10 ring road which engulfs Amsterdam. Figures 11.8 and 11.9 (produced by METAGRAF) depict examples of flow and speed trajectories, respectively, for the

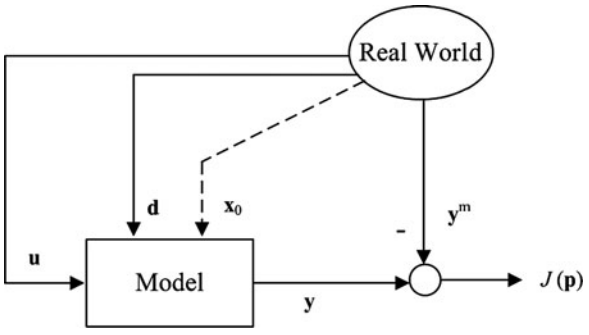


Fig. 11.5 Calculation of the performance criterion

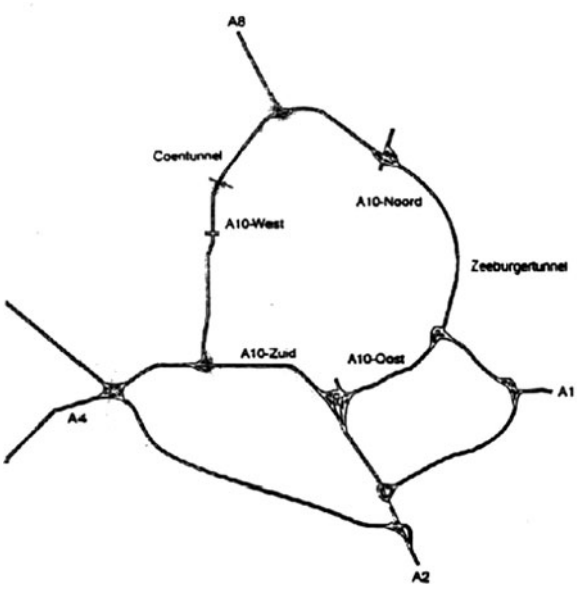


Fig. 11.6 Amsterdam motorway network

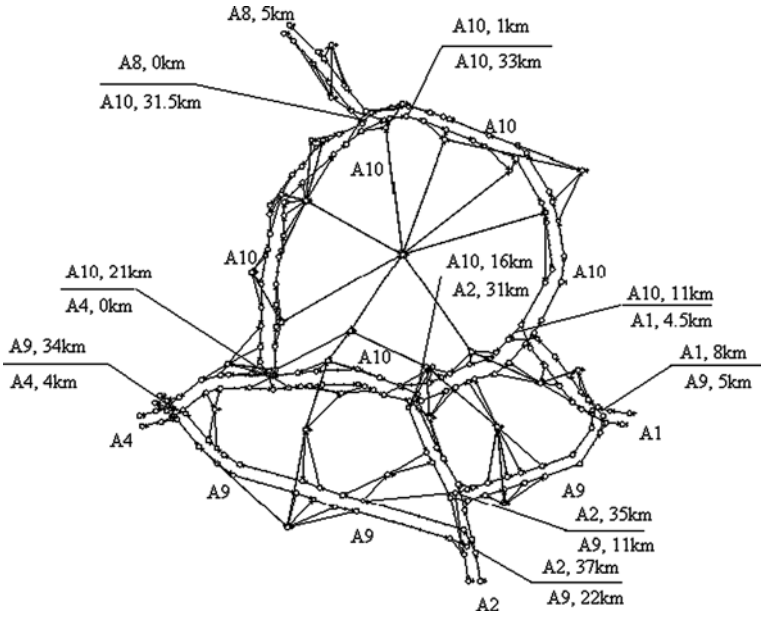


Fig. 11.7 Amsterdam network representation

link L11 of A10 clockwise direction, compared with real measurements taken from the same location. Figure 11.10 presents snapshots of the traffic conditions for the whole network at certain time instants. Free, dense, and congested conditions are present, and each segment is colored appropriately to indicate them while the links'

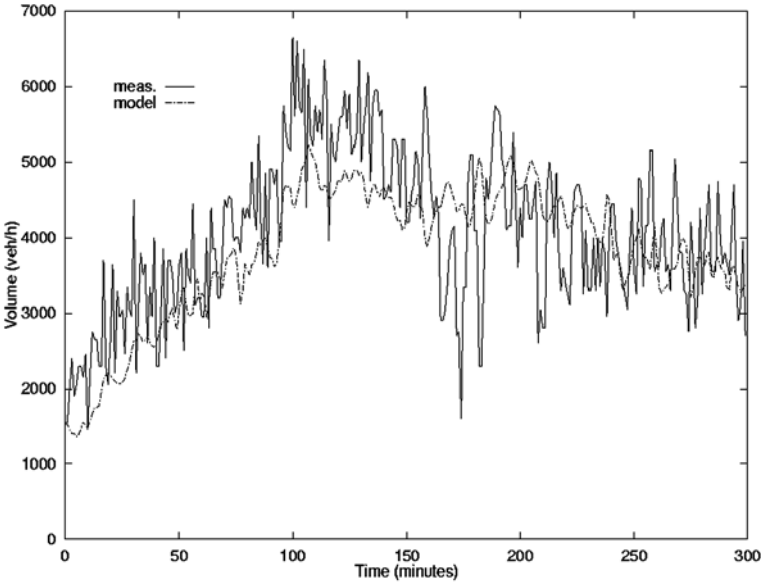


Fig. 11.8 Measured versus predicted flow

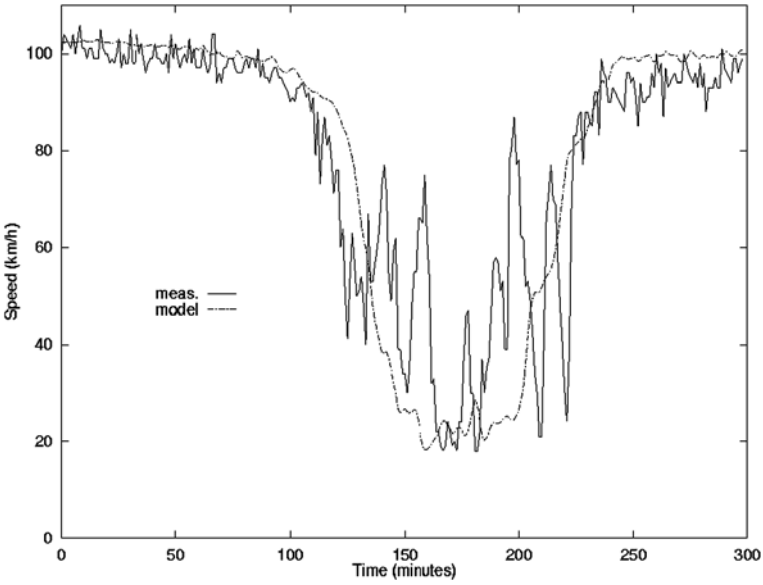


Fig. 11.9 Measured versus predicted speed

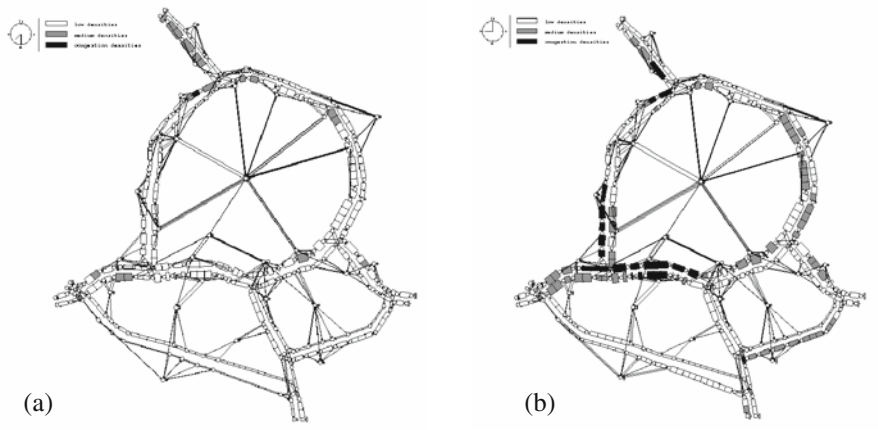


Fig. 11.10 Simulated traffic conditions in the network at 7:30 a.m. (**a**) and 9:00 a.m. (**b**)

segment width is proportional to the traffic flow passing through them. The model predicts the traffic conditions sufficiently accurately, which demonstrates that it is a suitable tool for evaluating the impact of various traffic control measures on the traffic flow process.

11.6 Extended Modeling Capabilities

The METANET model and software described in previous sections may be used for simulation studies of various kinds. On the other hand, thanks to the transparent (state-space form) macroscopic model (and related software), the tool provided an excellent basis for extensions toward the development of other generic model-based tools aiming at quite different endeavors, such as real-time decision support, exact dynamic traffic assignment, system optimal traffic assignment and routing, optimal traffic control encompassing a variety of control measures, and real-time traffic surveillance. This section outlines these METANET extensions.

11.6.1 Online Metanet

In view of the very low computation times needed for METANET in cases of large-scale motorway networks, the tool may be embedded in motorway traffic control centers for online use aiming at decision support for traffic operators. More specifically, a traffic operator facing a normal or abnormal (e.g., incident and special event) traffic situation may be interested to know in advance the impact of potential control measures (such as driver information or guidance and ramp metering) on the evolution of the current traffic conditions. In this context, online METANET

can be deployed to assess and compare various control alternatives before actually releasing the most promising one. Such a METANET-based system was actually developed for the motorway network of The Netherlands (Smulders et al., 1999; Knibbe 2000; Knibbe and Kock 2001). A similar METANET-based system was also developed by a French traffic operations company, albeit with control scenarios created by an expert system rather than a human operator (Morin et al., 1991).

In more technical terms, the functioning of online METANET may be described accurately on the basis of the state-space model (eq. (11.17)):

- The current traffic situation corresponds to the initial condition $\mathbf{x}_0$ which is provided by the corresponding measurements of the traffic control center; this may be done automatically (without operator intervention) if METANET is embedded in the control center.
- The control $\mathbf{u}(k)$ is provided by the operator, possibly out of a pre-selected set of control actions. A number of different control scenarios may be defined by the operator in this way.
- The disturbance vector $\mathbf{d}(k)$ may be based on an extrapolation of the prevailing boundary conditions over the required prediction horizon K or on related stored historical values.
- Finally the parameter vector $\mathbf{p}$ may have to be altered to reflect different characteristics in case of incidents.

Based on the above, the model (eq. (11.17)) is run over a future horizon K for each defined control scenario, delivering corresponding performance indexes and diagrams that would enable the operator to make a sensible decision or simply to decide whether a particular control action would be better than a do-nothing scenario.

11.6.2 Metanet-DTA

Section 11.4 outlined a feedback "one-shot" procedure for approximate dynamic traffic assignment in the sense of a dynamic user equilibrium. An alternative to feedback algorithms that can be used for dynamic traffic assignment or route guidance, is iterative procedures that may aim at establishing either user-optimal or system-optimal conditions. For the system-optimal case, the goal is the minimization of a global objective criterion (e.g., the total time spent) even for the price of partially following routes that are more costly than the regular routes and the corresponding procedure is outlined in Section 11.6.3. On the other hand, the typical structure of an iterative procedure toward exact user-optimal conditions is the following (Wang et al., 2001):

a. Set the initial splitting rates.
b. Run the simulation model (as in Fig. 11.4a) over a time horizon.

c. Evaluate the experienced (predictive) travel times on alternative routes; if all travel time differences are sufficiently small, stop with the final solution.
d. Modify the splitting rates appropriately to reduce travel time differences; go to (b).

The destination-oriented macroscopic dynamic model presented in Section 11.4 is used in step (b) according to Fig. 11.4a. The modification of the splitting rates in step (d) can be done using decentralized formulas like the Frank–Wolfe (Wisten and Smith 1997) or a PI formula (Wang et al., 2001).

The described iterative algorithm is employed by an extension of METANET called METANET-DTA. The METANET-DTA software may be used for exact dynamic user equilibrium for the price of increased computational time (compared to METANET) which increases by a factor roughly equal to the number of required iterations.

11.6.3 AMOC

Prevention or reduction of traffic congestion on motorway networks can dramatically improve the efficiency of the infrastructure in terms of throughput. Therefore, an increasingly important area in the field of traffic engineering is motorway traffic control. The macroscopic model and software of METANET has formed the basis for the generic open-loop optimal control tool AMOC (Advanced Motorway Optimal Control) first presented by Kotsialos et al. (2002b). AMOC is based on the macroscopic traffic flow model (eq. (11.17)) or its destination-oriented counterpart of Section 11.4, according to the choice of the user, and produces the control trajectories (over a time horizon K) that minimize the total time spent in the network subject to suitable control variable bounds. The selectable control inputs include ramp metering, splitting rates (for route guidance or system-optimal dynamic traffic assignment), motorway-to-motorway control, variable speed limits, and mainstream signal control, or any combination of these control measures, located at any selectable network location (integrated traffic control).

In more technical terms, the open-loop optimal control problem solved within AMOC reads: Given an initial state condition $\mathbf{x}_0$ and boundary variable trajectories $\mathbf{d}(k), k = 0, \ldots, K-1$, find the optimal control trajectory $\mathbf{u}^*(k), k = 0, \ldots, K-1$, that minimizes an objective function of the form

$$J = \vartheta[\mathbf{x}(K)] + \sum_{k=0}^{K-1} \varphi[\mathbf{x}(k), \mathbf{u}(k), \mathbf{d}(k)] \tag{11.26}$$

subject to eq. (11.17) and the inequality constraints imposed on the control variables. Here, ϑ and φ are arbitrary, twice differentiable, nonlinear cost functions.

The specific cost criterion considered in AMOC is the total time spent (TTS) of all vehicles in the network (including the waiting time experienced in the ramp queues) which is a natural objective for the traffic systems considered.

$$J = T \sum_{k=1}^{K-1} \sum_{m} \sum_{i} \rho_{m,i}(k) L_m \Lambda_m + T \sum_{k=1}^{K-1} \sum_{o} w_o(k) \tag{11.27}$$

As mentioned earlier, the control vector **u** may comprise any user-specific control measures at any network locations. If route guidance is involved, then the destination-oriented model version (Section 11.4) must be selected. If ramp metering measures are included, then maximum ramp queue constraints may be taken into account via the introduction of penalty terms in the cost criterion penalizing queue lengths larger than $w_{\max,o}$, which is a predetermined maximum admissible number of vehicles in the queue of origin o. Another penalty term may be added in order to suppress high-frequency time variations of the optimal control trajectories.

The described optimal control problem is automatically solved by AMOC within few minutes even for large-scale networks using a powerful gradient-based feasible direction algorithm (Papageorgiou and Marinaki 1995). The solution algorithm is iterative in nature whereby the control variables are modified at each iteration (based on the gradient of J) so as to reduce the value of the objective function J.

11.6.4 RENAISSANCE

Successful motorway traffic control and management calls for sufficient and reliable information on the current and, in some cases, on the short-term future traffic conditions. Related issues and problems include the following:

- sparse detectors in some network parts;
- faulty detectors;
- incidents that are not directly detected; and
- short-term prediction issues.

The macroscopic model and software of METANET (only the non-destination-oriented version) has formed the basis for the generic motorway surveillance tool RENAISSANCE (REal-time motorway Network trAffIc State SurveillANCE) (Wang et al., 2004, 2006) that addresses these and further issues.

Probably the most significant task of RENAISSANCE is the real-time estimation of the complete traffic state, based on a limited amount of sensor information, by the use of an extended Kalman filter (EKF). In more technical terms, any available measurements (typically flows and mean speeds) may be related to state variables via an output equation

$$\mathbf{y}(k) = \mathbf{g}[\mathbf{x}(k)] \tag{11.28}$$

where **y** is the vector of available measurements and **g** is a corresponding differentiable function. Although the number of available measurements (i.e., the dimension of the vector **y**) may be much higher than the system state (i.e., the dimension of the

vector $\mathbf{x}$), an EKF may be used to produce an estimate $\hat{\mathbf{x}}$ of the current system state via a combination of (a stochastic version of) model (eq. (11.17)) and the arriving measurements $\mathbf{y}(k)$:

$$\hat{\mathbf{x}}(k+1) = \mathbf{f}\left[\hat{\mathbf{x}}(k)\right] + \mathbf{K}(k)\left[\mathbf{y}(k) - \mathbf{g}\left[\hat{\mathbf{x}}(k)\right]\right] \tag{11.29}$$

where $\mathbf{K}$ is the EKF gain matrix that is calculated online appropriately. Note that all external variables and parameters included in the model (eq. (11.17)) are rendered additional state variables in eq. (11.29) via the introduction of additional random-walk state equations, see (Wang et al., 2004, 2006) for details.

Beyond traffic state estimation, RENAISSANCE delivers in a real-time environment:

- short-term traffic state prediction;
- model parameter estimation;
- automatic incident detection;
- current and short-term future congested areas; and
- travel time prediction for en-route driver information.

RENAISSANCE is a generic tool that is applicable in real time to motorway networks of arbitrary size, topology, and characteristics, based on any suitable traffic detector configuration; it can be easily integrated in traffic control centers to enhance and extent the available real-time information for various uses.

11.7 Selected Overview of Advanced Case Studies and Applications

This section presents some selected applications of METANET and AMOC.

11.7.1 Automatic Control of VMS in the Interurban Scottish Highway Network

The design, implementation, and field evaluation work performed within the European DRIVE II project QUO VADIS for the development of VMS information and guidance system in the interurban Scottish highway network (Fig. 11.11) based on simple automatic control methodologies was presented by Messmer et al. (1998). Simulation studies performed prior to the field implementation using METANET demonstrated the potential improvements. The model used by METANET was previously validated using real measurements. Figure 11.12 (produced by METAGRAF) depicts the southbound part of the Scottish highway network as considered by METANET. The simulation runs addressed the morning

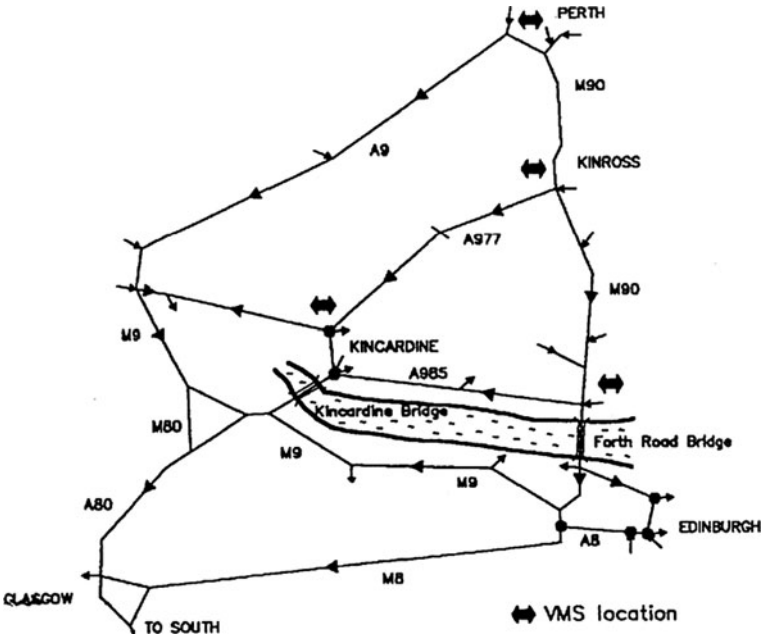


Fig. 11.11 The Scottish interurban network

peak of a typical working day. A number of different scenarios were investigated assuming incidents at different key locations in the network. As an example, Fig. 11.13 presents a snapshot of the traffic conditions for the whole network at 8:30 a.m. when an accident is present on Forth Road Bridge and VMS automatic control is applied. White areas in the links correspond to undercritical traffic densities, while grey and black areas correspond to critical and congested conditions, respectively. The links' segment width is proportional to the traffic flow passing through them.

11.7.2 Ramp Metering Pilot Project for the Monash Freeway

VicRoads conducted within 2008 a coordinated ramp metering pilot project on six on-ramps of the Monash Freeway (Melbourne, Australia) extending from Jacksons RD to Warrigal RD (Fig. 11.14). The pilot project implemented successfully the ALINEA/HERO coordination scheme (Papamichail and Papageorgiou 2008) for the six metered ramps. A detailed study was conducted prior to the field implementation (Papamichail et al., 2007) that aimed to address the following main issues:

- Realistic modeling of the traffic flow on the pilot project motorway stretch using METANET;

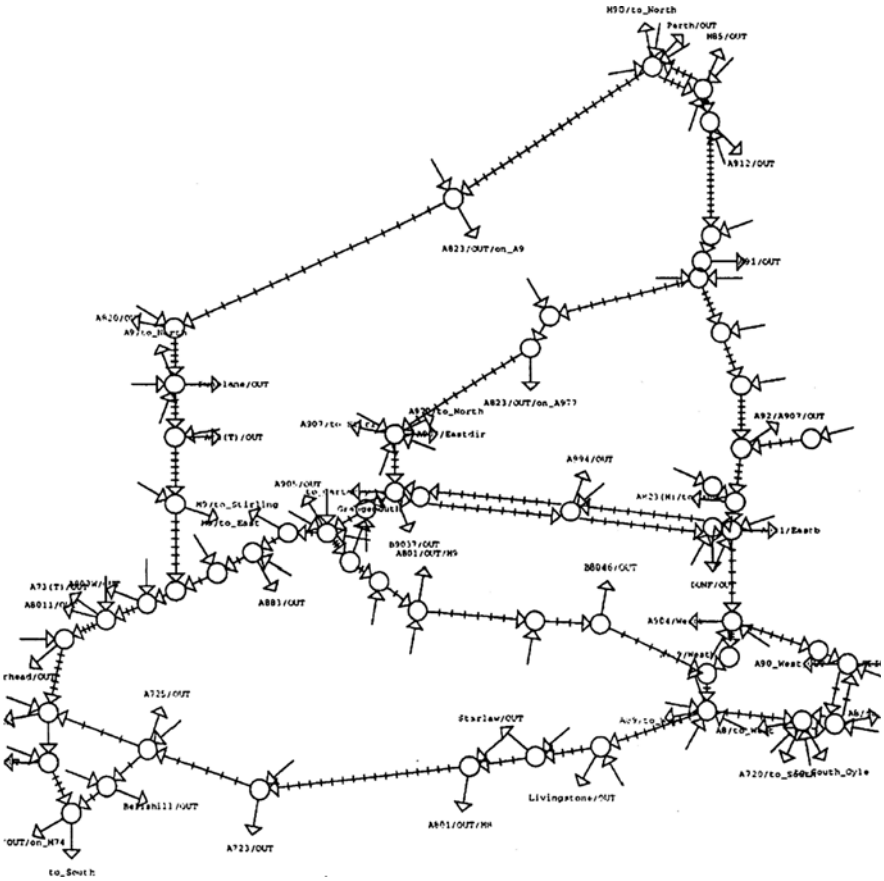


Fig. 11.12 Representation of the Scottish interurban network in METANET

- Appropriate configuration of the generic ramp metering ALINEA/HERO control software for the specific application;
- Simulation-based investigation of the potential traffic improvements via ramp metering under various scenarios.

A number of work phases were carried out as summarized in the following:

- Static data and dynamic data collected from all detector stations of the considered pilot area were processed appropriately for use in METANET.
- METANET was applied for simulation of the pilot stretch (Fig. 11.15) and was validated by use of real traffic data according to Section 11.5. After calibration of the model parameters, the simulator was found to reproduce the spatio-temporal traffic evolution within the considered motorway part with sufficient accuracy

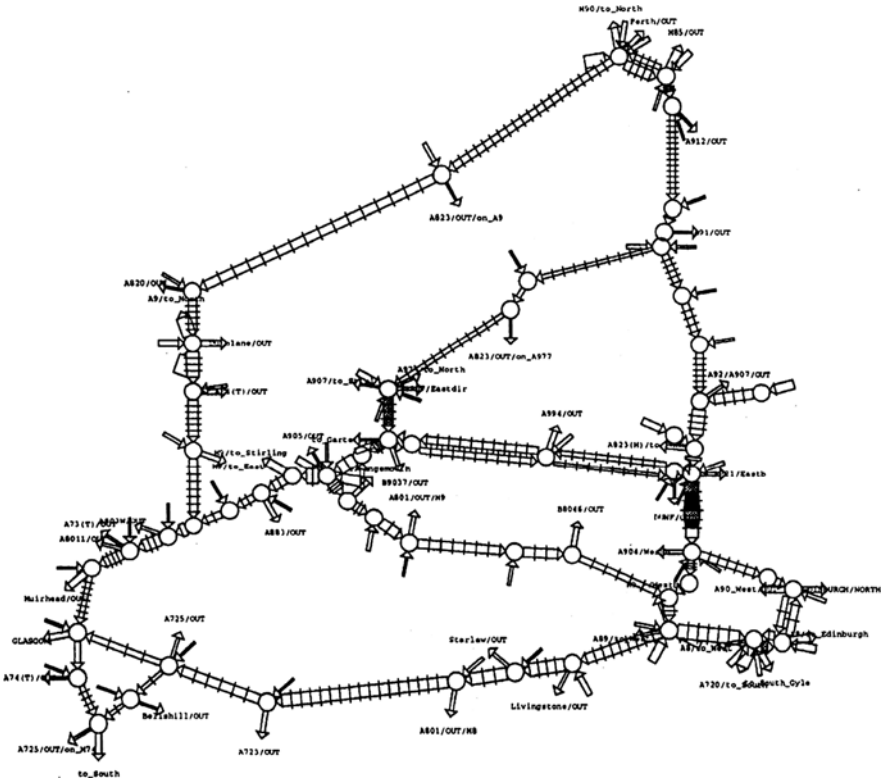


Fig. 11.13 Simulated accident on Forth Road Bridge: VMS automatic control

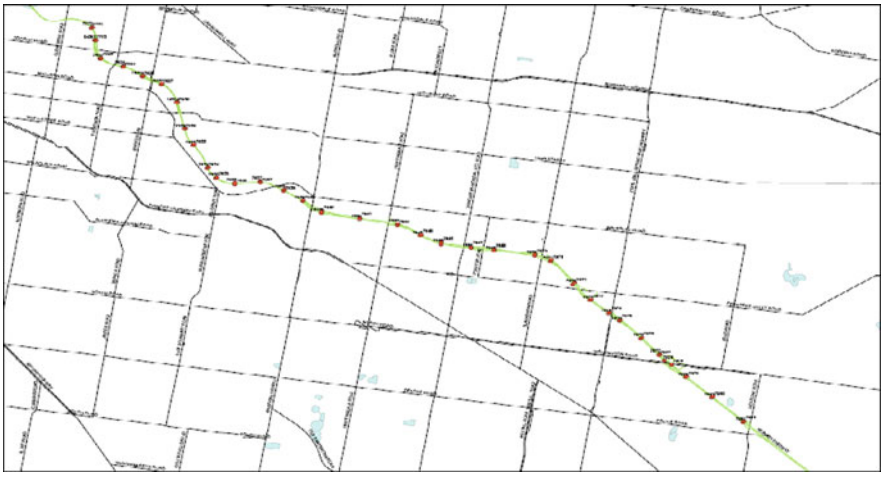


Fig. 11.14 The Monash Freeway pilot project stretch

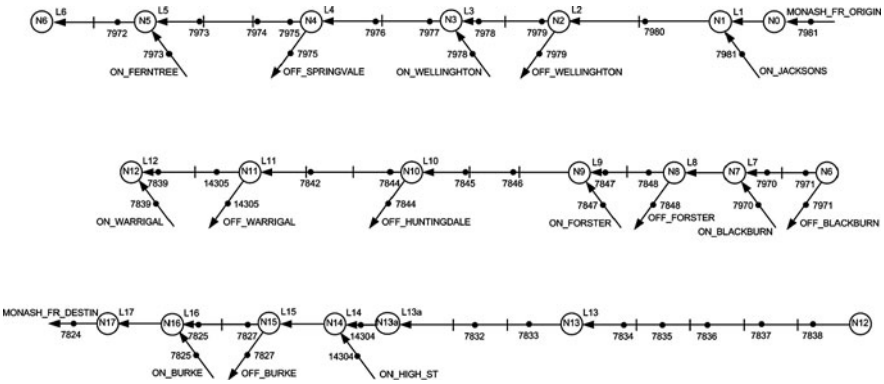


Fig. 11.15 Representation of the Monash Freeway pilot project stretch in METANET

when fed with measured boundary data. Figures 11.16 and 11.17 depict measured versus predicted speed and flow data for specific locations distributed along the motorway stretch. Mainstream measured data appear in the figures with blue curves and the detector station id is given for each trajectory. The corresponding predicted trajectories appear with red curves. The link name followed by the segment number is specified for each one of the trajectories.

- The ALINEA/HERO control software was configured for application to the pilot project area. The control software was interconnected with METANET according to Section 11.3.4 in order to emulate the real closed-loop-controlled system behavior under a variety of cases and choices that are reflected in a number of corresponding scenarios tested.

11.7.3 Optimal Control Results for the Amsterdam Ring Road

Optimal control results obtained using AMOC have been utilized (Kotsialos and Papageorgiou 2004; Papamichail et al., 2010; Carlson et al., 2010) for studying the application of several control measures to the counter-clockwise direction of the Amsterdam A10 ring road (Fig. 11.18).

Figure 11.19 shows the traffic density (in veh/km/lane) space–time profiles for the Amsterdam ring road under the following conditions:

(a) No control;
(b) Coordinated ramp metering with storage space on the urban on-ramps for 30 vehicles leads to 22% improvement in TTS in the network compared to the no-control case;
(c) VSL lead to 47% improvement in TTS compared to the no-control case; and

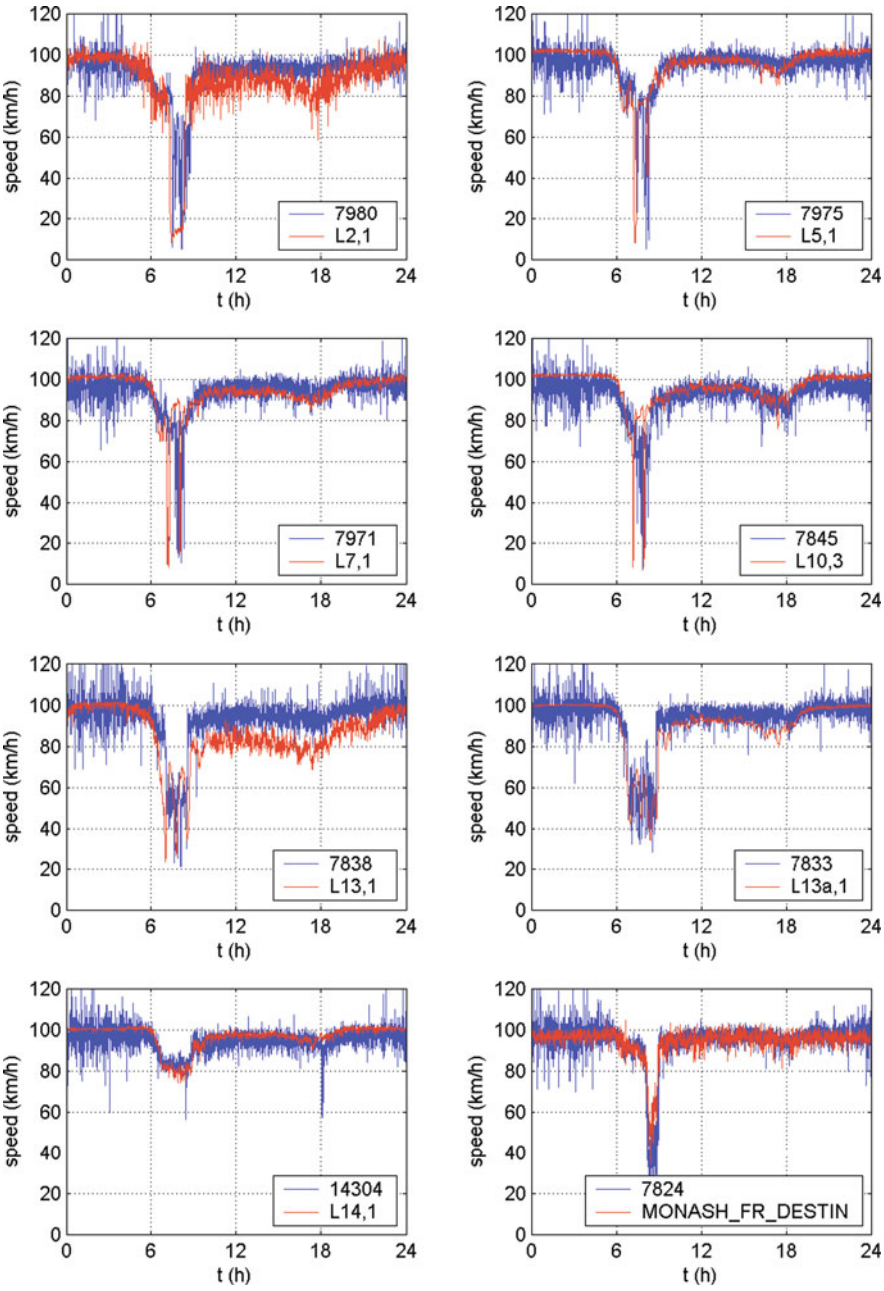


Fig. 11.16 Measured versus predicted speed for 10.08.2006

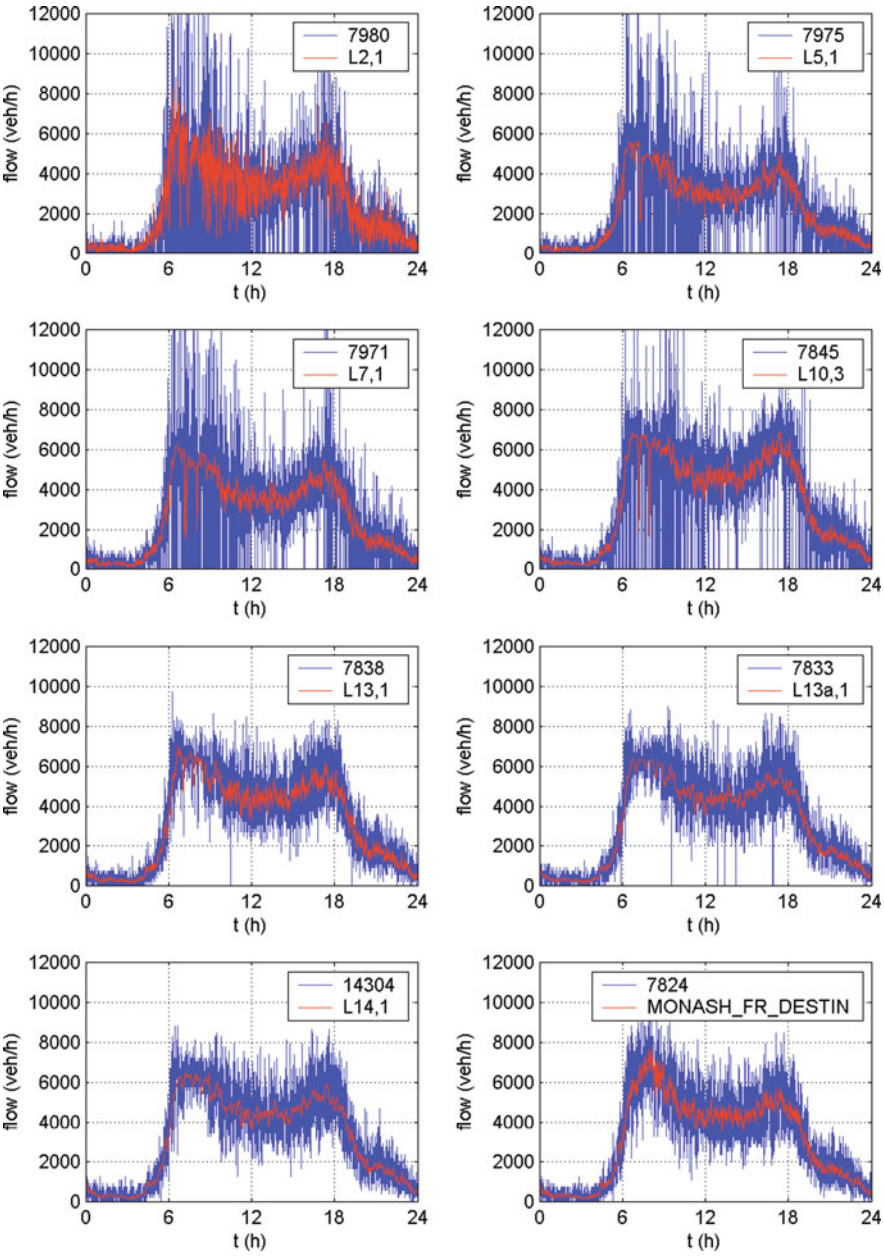


Fig. 11.17 Measured versus predicted flow for 10.08.2006

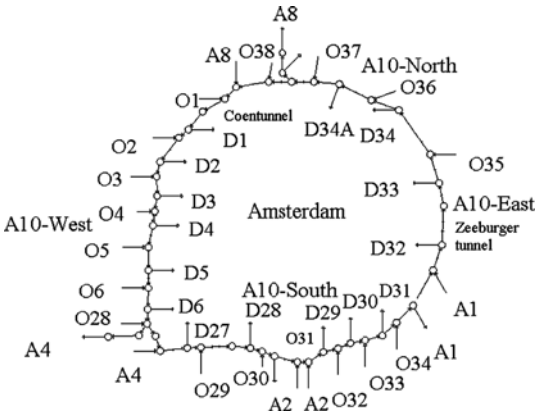
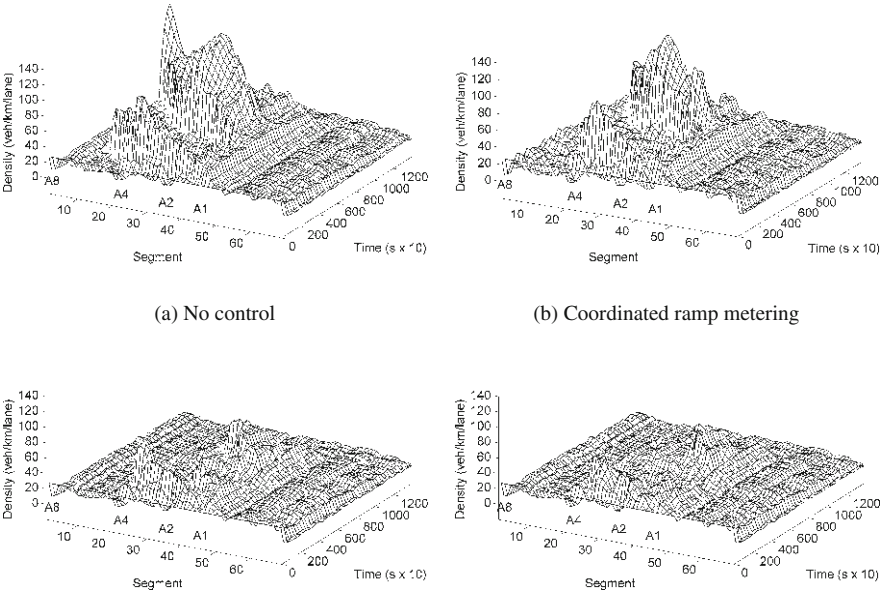


Fig. 11.18 The Amsterdam A10 ring road (counter-clockwise direction)



(a) No control

(b) Coordinated ramp metering

(c) VSL

(d) VSL and coordinated ramp metering

Fig. 11.19 Density space–time profiles for the Amsterdam ring road

(d) VSL and coordinated ramp metering with storage space on the urban on-ramps for 30 vehicles lead to 49% improvement in TTS compared to the no-control case.

Note that the model used by METANET for the no-control case and by AMOC for the optimal control results was the one validated by Kotsialos et al., (2002a), see

Section 11.5. For the no-control case (Fig. 11.19a) the excessive demand, coupled with the uncontrolled entrance of the drivers into the mainstream, causes congestion shortly after the beginning of the simulation period. This congestion originates at the junction between the A1 with the A10 (ring road) and propagates upstream, blocking A4 and a large part of the A10-West. After this congestion is partially dissolved, a new one appears and propagates upstream until it reaches the first congestion whose trend of resolving is thereby reversed leading to a single more severe congestion. This strong congestion spills back onto the A4 motorway creating a congestion of several kilometers there as well. For the ramp metering case (Fig. 11.19b), even with perfect knowledge of the demands, congestion is reduced but cannot be avoided (due to limited ramp storage space). For the VSL case (Fig. 11.19c) there are again two congestion bodies forming, but, in contrast to the no-control case, these congestion occurrences are much less extended in space and time. In fact, these controlled dense-flow areas are not the direct result of the bottleneck in the merge area of the A1 with the A10 (as in the no-control case), but due to holding back of traffic upstream of the A1/A10 merging bottleneck via appropriate (optimal) VSL. For the integrated control case (Fig. 11.19d), i.e., VSL and coordinated ramp metering, the mainline (controlled) congestion is even weaker than in the VSL case (Fig. 11.19c) because, roughly speaking, the vehicles stored in the on-ramps are taken out of the mainline congestion.

Due to various uncertainties, the open-loop optimal solution delivered by optimal control approaches becomes suboptimal when directly applied to the motorway traffic process, see, e.g., Papamichail et al., (2010). However, the optimal results can be used in a rolling horizon mode or can be utilized to extract useful conclusions for the development of similarly efficient but simpler feedback control strategies.

References

Carlson RC, Papamichail I, Papageorgiou M, Messmer A (2010) Optimal mainstream traffic flow control of large scale motorway networks. Transp Res Part C, 18:193–212

Cremer M, Papageorgiou M (1981) Parameter identification for a traffic flow model. Automatica, 17:837–843

Diakaki C, Papageorgiou M (1996) Integrated modeling and control of corridor traffic networks using METACOR modeling tool. Internal report 1996–1997, Dynamic Systems and Simulation Laboratory, Technical University of Crete, Chania, Greece

Elloumi N, Haj-Salem H, Papageorgiou M (1994) METACOR: a macroscopic modeling toll for urban corridor. TRISTAN – (Triennial Symposium on Transportation Analysis) II, Capri, Italy, vol 1, p 135–150

Hoogendoorn SP, Bovy PHL (2001) State-of-the-art of vehicular traffic flow modelling. Proc Inst Mech Eng, 215:283–303

Knibbe WJJ (2000) Macroscopic traffic simulation for on-line forecasting. Preprints 9th IFAC symposium on control in transportation systems, Braunschweig, Germany

Knibbe WJJ, Kock R (2001) Experimenting with decision support for traffic management. Proceedings of IEEE conference on transportation systems, Oakland, California, p. 1024–1028

Kotsialos A, Papageorgiou M (2004) Efficiency and equity properties of freeway network-wide ramp metering with AMOC. Transp Res Part C 12:401–420

Kotsialos A, Papageorgiou M, Diakaki C, Pavlis Y, Middelham F (2002a) Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool METANET. IEEE Trans Intell Transp Sys 3:282–292

Kotsialos A, Papageorgiou M, Mangeas M, Haj-Salem H (2002b) Coordinated and integrated control of motorway networks via nonlinear optimal control. Transp Res Part C 10:65–84

Lighthill MJ, Whitham JB (1955) On kinematic waves II: A theory of traffic flow on long crowded roads. Proc Royal Soc A, 229:317–345

Messmer A, Papageorgiou M (1990) METANET: a macroscopic simulation program for motorway networks. Traffic Eng Control, 31:466–470

Messmer A, Papageorgiou M (1994) Automatic control methods applied to freeway network traffic. Automatica, 30:691–702

Messmer A, Papageorgiou M, Mackenzie N (1998) Automatic control of variable message signs in the interurban Scottish highway network. Transp Res Part C 6:173–187

Morin JM, Morin Gower P, Papageorgiou M, Messmer A (1991) Motorway networks modelling and control. Proceedings of DRIVE Conference, Brussels, Belgium, p. 148–173

Papageorgiou M (1990) Dynamic modeling, assignment, and route guidance in traffic networks. Transp Res Part B 24:471–495

Papageorgiou M, Blosseville JM, Haj-Salem H (1989) Macroscopic modelling of traffic flow on the Boulevard Périphérique in Paris. Transp Res Part B 23:29–47

Papageorgiou M, Blosseville JM, Haj-Salem H (1990) Modelling and real-time control of traffic flow on the southern part of Boulevard Périphérique in Paris – Part I: modeling. Transp Res Part A 24:345–359

Papageorgiou M, Haj-Salem H, Middelham F (1997) ALINEA local ramp metering: summary of field results. Transp Res Rec 1603:90–98

Papageorgiou M, Marinaki M (1995) A feasible direction algorithm for the numerical solution of optimal control problems. Internal report 1995–1994, Dynamic Systems and Simulation Laboratory, Technical University of Crete, Chania, Greece

Papamichail I, Kampitaki K, Papageorgiou M, Messmer A (2008) Integrated ramp metering and variable speed limit control of motorway traffic flow. Proceedings 17th IFAC world congress (IFAC'08), Seoul, Korea, p. 14084–14089

Papamichail I. Kotsialos A, Margonis I, Papageorgiou M (2010) Coordinated ramp metering for freeway networks – A model-predictive hierarchical control approach. Transp Res Part C, 18:311–331

Papamichail I, Papageorgiou M (2008) Traffic-responsive linked ramp-metering control. IEEE Trans Intell Transp Sys 9:111–121

Papamichail I, Papageorgiou M, Kosmatopoulos E. (2007) Modelling, configuration and simulation testing for the VicRoads ramp metering pilot project – Final report. Prepared for VicRoads, Melbourne, Victoria, Australia

Payne HJ (1971) Models of freeway traffic and control. Simulation Counc Proc 1:51–61

Smulders SA, Messmer A, Knibbe WJJ (1999) Real-time application of METANET in traffic management centres. 6th world congress on ITS, Toronto, Canada

Wang Y, Messmer A, Papageorgiou M (2001) Freeway network simulation and dynamic traffic assignment with METANET tools. Transp Res Rec 1776:178–188

Wang Y, Papageorgiou M, Messmer A (2004) RENAISSANCE – A real-time motorway network traffic state surveillance tool, User manual. Dynamic Systems and Simulation Laboratory, Technical University of Crete, Chania, Greece

Wang Y, Papageorgiou M, Messmer A (2006) RENAISSANCE – A unified macroscopic model-based approach to real-time freeway network traffic surveillance. Transp Res Part C 14: 190–212

Wardrop JG (1952) Some theoretical aspects of road traffic research, Proc Inst Civil Eng 1:325–362

Wisten MB, Smith MJ (1997) Distributed computation of dynamic traffic equilibria. Transp Res Part C 5:77–93

Subject Index

J. Barceló (ed.), *Fundamentals of Traffic Simulation*,
International Series in Operations Research & Management Science 145,
DOI 10.1007/978-1-4419-6142-6, © Springer Science+Business Media, LLC 2010