Linköping Studies in Science and Technology. Dissertations
No. 1033

Design and Analysis of Diagnosis Systems Using Structural Methods

Mattias Krysander

Department of Electrical Engineering
Linköping University, SE–581 83 Linköping, Sweden

Linköping 2006

Design and Analysis of Diagnosis Systems Using Structural Methods

`matkr@isy.liu.se`
`http://www.fs.isy.liu.se`
Department of Electrical Engineering,
Linköpings universitet,
SE–581 83 Linköping,
Sweden.

ISBN 91-85523-32-1 ISSN 0345-7524

Printed by LiU-Tryck, Linköping, Sweden 2006

To my dear wife Åsa

Abstract

In complex and automated technological processes the effects of a fault can quickly propagate and lead to degradation of process performance or even worse to a catastrophic failure. This means that faults have to be found as quickly as possible and decisions have to be made to stop the propagation of their effects and to minimize process performance degradation. The behavior of the process is affected in different ways by different faults and the fault can be found by ruling out faults for which the expected behavior of the process is not consistent with the observed behavior. In model-based diagnosis, a model describes the expected behavior of the process for the different faults.

A device for finding faults is called a diagnosis system. In the diagnosis systems considered here, a number of tests check the consistency of different parts of the model, by using observations of the process. To be able to identify which fault that has occurred, the set of tests that is used must be carefully selected. Furthermore, to reduce the on-line computational cost of running the diagnosis system and to minimize the in general difficult and time-consuming work of tests construction, it is also desirable to use few tests.

A two step design procedure for construction of a diagnosis systems is proposed and it provides the means for selecting which tests to use implicitly by selecting which parts of the model that should be tested with each test. Then, the test design for each part can be done with any existing technique for model-based diagnosis.

Two different types of design goals concerning the capability of distinguishing faults is proposed. The first goal is to design a sound and complete diagnosis system, i.e., a diagnosis system with the following property. For any observation, the diagnosis system computes exactly the faults that together with the observation are consistent with the model. The second goal is specified by which faults that should be distinguished from other faults, and this is called the desired isolability.

Given any of these two design goals, theory and algorithms for selecting a minimum cardinality set of parts of the model are presented. Only parts with redundancy can be used for test construction and a key result is that there exists a sound and complete diagnosis system based on the set of all minimal parts with redundancy in the model. In differential-algebraic models, it is in general difficult to analytically identify parts with redundancy, because it corresponds to variable elimination or projection. It is formally shown that redundant parts can be found by using a structural approach, i.e., to use only which variables that are included in each equation. In the structural approach, parts with more equations than unknowns are identified with efficient graph-theoretical tools. A key contribution is a new algorithm for finding all minimal parts with redundancy of the model. The efficiency of the algorithm is demonstrated on a truck engine model and compared to the computational complexity of previous algorithms.

In conclusion, tools for test selection have been developed. The selection is based on intuitive requirements such as soundness or isolability requirements specified by the diagnosis system designer. This leads to a more straightforward design of diagnosis systems, valuable engineering time can be saved, and the resulting diagnosis systems use minimum number of tests, i.e., the on-line computational complexity of the resulting diagnosis systems become low.

Acknowledgments

This work was performed at the department of Electrical Engineering, division of Vehicular Systems, Linköping University. I am grateful to professor Lars Nielsen for letting me join this group and for supporting me during this work.

Foremost, I would like to thank my supervisor, Dr. Mattias Nyberg, and express my sincere gratitude for his guidance, encouragement, and for always taking time for discussions. I would also like to thank him for proofreading the manuscripts and giving me suggestions for improvements.

I am grateful to Dr. Erik Frisk and Dr. Jan Åslund for proofreading this thesis. They have given valuable comments on my work.

Furthermore, I would like to thank Erik Frisk and Jonas Biteus for helping me out with LaTeX issues. I would also like to thank Carolina Fröberg and Susana Högne for helping me with all sorts of things and to the staff at Vehicular Systems for creating an excellent atmosphere.

The financial support by the Swedish Agency for Innovation Systems (VINNOVA) through the center of excellence Information Systems for Industrial Control and Supervision (ISIS) is gratefully acknowledged. I would like to thank Dr. Inger Klein for the work involving ISIS.

Finally, I would like to thank my dear wife, Åsa, and my family for their support and patience.

Linköping, June 2006

Mattias Krysander

C

Notation 1

1 Introduction 5

1.1 Basic Principles of Fault Detection and Isolation 6
1.2 Basic Principles for Consistency Checking 6
1.3 Architecture of a Diagnosis System 7
1.3.1 Good Fault Isolation Capability 8
1.4 Diagnosis System Construction 8
1.4.1 Successful Test Construction 9
1.4.2 Objectives of Model Selection 9
1.4.3 Finding Model Redundancy Using Structural Methods . 10
1.5 Main Objective 10
1.6 Summary and Contribution of the Thesis 11
1.7 The Results from a User Perspective 15
1.7.1 Deriving Fault Isolability Requirements 16
1.7.2 Constructing a Diagnosis Model of the Process 16
1.7.3 Performing Detectability and Isolability Analysis 16
1.7.4 Selecting Models for Test Construction 17
1.7.5 Designing Tests for the Selected Models 17
1.7.6 Selecting Fault Isolation Algorithm 17
1.8 Main Contributions 18
1.9 Publications 19

2 A Framework for Model Based Diagnosis 21

2.1 Modeling Framework 22
2.1.1 Components 22
2.1.2 Component Behavioral Modes 23
2.1.3 Component Behavior 23
2.1.4 System and System Behavioral Modes 24

2.1.5 Diagnosis . . . 26

2.2 Diagnosis Tests . . . 27

2.2.1 Diagnosis Tests and the Model . . . 28

2.3 Diagnosis System . . . 30

2.3.1 Strategies for Designing Diagnosis Systems . . . 30

2.3.2 Starting From Null Hypotheses . . . 30

2.3.3 Starting From the Model . . . 31

2.4 Connection to FDI Methods . . . 33

2.5 Completeness and Soundness Properties of Diagnosis Systems . 35

2.5.1 Designing a Complete Diagnosis System . . . 36

2.5.2 Designing a Sound Diagnosis System . . . 37

2.6 Testing Small Models . . . 39

2.7 Systems with Noise . . . 41

2.8 Conclusions . . . 43

3 Soundness when Diagnosing Linear Static Systems 45

3.1 Linear Static Models . . . 46

3.1.1 Observation Sets . . . 47

3.1.2 Redundancy and Rejectable Models . . . 48

3.2 A Proper Overdetermined Set is Sufficient for Soundness . . . 49

3.3 Properties of PO sets . . . 50

3.4 Minimal Overdetermined Sets are Sufficient for Soundness . . . 53

3.4.1 Defining and characterizing MO sets . . . 54

3.4.2 Motivating Example for Using MO Sets . . . 54

3.4.3 Observation Set for a Family of Equation Sets . . . 55

3.4.4 Redundancy of a Family of Equation Sets . . . 56

3.4.5 Determine Consistency of Models by Testing MO Subsets 57

3.5 Selecting MO Sets . . . 61

3.6 Linear Models with Constants . . . 63

3.7 Redundancy and the Number of MO Sets . . . 63

3.7.1 Redundancy Dependence of the Number of Sensors . . . 64

3.7.2 The Number of MO Sets . . . 64

3.8 Diagnosing all Behavioral Models . . . 65

3.8.1 Sufficient to Test All MO Sets . . . 65

3.9 Minimizing The Number of Tests . . . 66

3.9.1 Algorithm . . . 66

3.9.2 Examples . . . 69

3.10 Fault Influence on Residuals . . . 72

3.10.1 Fault Influence in Models with Fault Variables . . . 73

3.10.2 Fault Influence in General Linear Static Models . . . 76

3.10.3 Fault Influence and Null Hypothesis . . . 77

3.11 Conclusions . . . 80

4 An Efficient Algorithm for Finding all MO Sets 83

4.1 Introduction . . . 83

4.2 The Proper Overdetermined Part . . . 84

4.2.1 Existence and Uniqueness . . . 84

4.2.2 Computing the PO Part . . . 86

4.3 Algorithm . . . 88

4.3.1 Input to the Algorithm . . . 88

4.3.2 Basic Algorithm 88
4.4 Improvements 90
4.4.1 A Canonical Partition of PO Sets 90
4.4.2 Analytical Reduction 92
4.4.3 Improved Algorithm 95
4.4.4 Algorithm without Lumping 98
4.5 Computational Complexity 99
4.6 Example: The Electrical Circuit 99
4.7 Conclusions 101

5 An Algorithm for Finding all Feasible MO Sets 103
5.1 Relation between Inputs and Outputs of Algorithm 4 104
5.2 Constructing Partitions by Using the Model 106
5.3 Computing the set M_{ne} 109
5.4 Generating Feasible Subsets 112
5.5 Algorithm for Finding All Feasible MO Sets 113
5.6 Computational Complexity 115
5.7 Conclusions 116

6 Soundness when Diagnosing Linear Dynamic Systems 117
6.1 Linear Differential Equations 118
6.1.1 Behavior 118
6.1.2 Observation Sets 119
6.1.3 Kernel Representation 120
6.1.4 Redundancy and Overdetermined Sets 122
6.2 PO Sets are Sufficient for Soundness 122
6.3 MO Sets are Sufficient for Soundness 124
6.3.1 Defining and Characterizing MO sets 124
6.3.2 Rank Condition 124
6.3.3 All MO Sets are Sufficient for Soundness 128
6.4 Diagnosing all Behavioral Models 131
6.4.1 Sufficient to Test All MO Sets 131
6.5 Algorithm for Finding All MO Sets in a Linear Dynamic Model 131
6.6 Minimizing The Number of Tests 132
6.7 Fault Influence on Residuals 134
6.7.1 Fault Influence and Null Hypothesis 134
6.8 Conclusions 136
6.A Some Properties of Polynomial Matrices 138
6.A.1 Polynomial Basis 139

7 Soundness when Diagnosing General Systems 141
7.1 Model Definitions 141
7.2 All Behavioral Models Sufficient for Soundness 143
7.3 All Minimal Rejectable Models are Sufficient for Soundness . . 144
7.4 A Sufficient and Necessary Condition for Soundness 146
7.5 Conclusions 149

8 Finding Rejectable Models Using Structural Methods 151
8.1 Some Basic Graph Theoretical Concepts 152
8.1.1 Matchings 153
8.2 Structured Matrices 153
8.2.1 Structured Matrix Representations 154
8.2.2 Rank of Structured Matrices 155
8.2.3 Graph-theoretical Characterization of Structural Rank . 155
8.2.4 General Matrix Rank Assumption 157
8.3 Structural Model Properties 158
8.4 Structural Conditions for Soundness 161
8.5 Structural Representations of Non-linear Models 162
8.5.1 The Structure of Dynamic Models 163
8.6 Finding Rejectable Models by Using Structural Methods 165
8.6.1 Structural vs Analytical Redundancy 165
8.6.2 Test based on MSO Sets 166
8.6.3 Relations to Other Structural Characterizations 167
8.7 Conclusions 168

9 An Efficient Algorithm for Finding all MSO sets 169
9.1 Introduction to the New Algorithm 170
9.1.1 Structural Assumption 170
9.1.2 Computing the PSO Set with Maximum Redundancy . . 171
9.2 Theoretical Foundation 173
9.2.1 Surplus of Equations 173
9.2.2 Characterizing Structural Redundancy 174
9.2.3 Characterizing the PSO Part M^+ 174
9.2.4 Characterizing PSO Sets 174
9.2.5 PSO Part and PSO Set 175
9.3 New Algorithm 176
9.4 Improvements 178
9.4.1 Structural Reduction 178
9.4.2 Improved Algorithm 181
9.5 Computational Complexity 184
9.6 Comparison with Previous Algorithms 185
9.6.1 Previous Algorithms 185
9.6.2 Computational Complexity Comparison 189
9.6.3 Worst Case Analysis 189
9.7 Application to a Large Industrial Example 190
9.8 Conclusions 191

10 Structural Analysis of MSO Sets of Differential Algebraic Equations 193
10.1 Differentiated MSO Set 194
10.2 Use of Differentiated MSO Sets 195
10.2.1 Differentiated MSO Set for Test Construction 195
10.2.2 Differentiated MSO Sets for Prediction 197
10.3 Structural Differentiation 198
10.4 Algorithm for Computing a Differentiated MSO Set 200
10.5 Basic Definitions and Notation 201
10.6 Existence of a Differentiated MSS Set 203
10.7 Uniqueness of Differentiated MSS Sets 205

10.8 Conclusions 212

11 Test Selection Based on Isolability Analysis of Diagnosis Systems 213

11.1 Analytical Isolation Capability 214
11.2 Problem Formulation 216
11.3 Predicting the Isolability 216
11.4 Computing Structural Isolability 218
11.5 Isolability Properties and Representations 219
11.6 Analytical Isolability Properties 221
11.7 Comparison between Structural and Analytical Isolability . . . 222
11.8 Sufficient Condition for Equality 225
11.9 Test Selection Based on Isolability Analysis 227
11.9.1 Necessary Condition for Desired Isolability 227
11.9.2 Test Selection 228
11.9.3 Computing Missing Isolability Properties 228
11.9.4 Obtaining Maximum Isolability 229
11.9.5 Example 230
11.10Conclusions 231

12 Isolability Prediction of Diagnosis Models and its Applications 233

12.1 Isolability of a Diagnosis Model 234
12.1.1 Predicting Detectability 235
12.1.2 Predicting Isolability 239
12.1.3 Isolability and Checking Models 240
12.2 Finding Checking Models by Using PSO Parts 241
12.3 Finding Checking Models by Removing Equations 244
12.3.1 Excluding Non-monitorable Equations 244
12.3.2 Structural Method 245
12.3.3 Algorithm 246
12.4 Computation of Isolability Prediction 248
12.4.1 Isolability Prediction Interpretation 252
12.5 Modeling Guidelines for Good Isolability Predictions 252
12.5.1 Motivation of the Guidelines 254
12.5.2 Fulfilling Guideline (a) 256
12.5.3 Fulfilling Guideline (b) 257
12.6 Applications of Isolability Prediction 258
12.6.1 Fault Modeling Specification 258
12.6.2 Design Alternative Evaluation 259
12.7 Designing a Diagnosis System with Maximum Isolability 259
12.7.1 Characterizing Maximum Isolability 260
12.7.2 Lower and Upper Bounds for the Maximum Isolability . 261
12.7.3 Selecting the Set of Potential Tests Δ_a 262
12.7.4 Design Procedure 263
12.8 Conclusions 264

References 267

N

Diagnosis Model Notation

$\mathbb{M}$	A diagnosis model.
$\mathbf{B}$	The set of all system behavioral modes in a diagnosis model.
$\mathbf{C}$	The set of all components in a diagnosis model.
$\boldsymbol{M}$	The set of all equations in a diagnosis model.
$\boldsymbol{X}$	The set of all unknown variables in a diagnosis model.
$\boldsymbol{Z}$	The set of all known variables in a diagnosis model.
$\mathbb{X}$	The domain of the unknown variables $\boldsymbol{X}$.
$\mathbb{Z}$	The domain of the known variables $\boldsymbol{Z}$.
$\boldsymbol{x}$	A vector of the unknown variables in $\boldsymbol{X}$.
$\boldsymbol{z}$	A vector of the known variables in $\boldsymbol{Z}$.
$\boldsymbol{f}$	A vector of fault variables.
assump(M)	A set of system behavioral modes that infer all equations in a set M.
$\mathcal{D}(\boldsymbol{z})$	The set of diagnoses when observing $\boldsymbol{z}$.
$\mathbf{B}_c$	The set of component behavioral modes defined for a component c.
M_b	The behavioral model for a system behavioral mode b.
M_c	The model that describes the behavior of a component c.
$M_{c,b}$	A behavioral model for a component behavioral mode b.
$\mathcal{B}_c(b)$	The external behavior for a component behavioral mode b.
sys	The system behavioral mode that the process is in.

Model Notation

E_d	A differentiated MSO set.
X_d	The unknowns in a differentiated MSO set.
$\bar{\boldsymbol{X}}$	The set of all time-derivatives of all unknown variables.
$e^{(k)}$	The k:th time-derivative of an equation e.
$x^{(k)}$	The k:th time-derivative of a variable x.
$E^{(k)}$	The set $\{e^{(k)} \vert e \in E\}$ of equations.
$X^{(k)}$	The set $\{x^{(k)} \vert x \in X\}$ of equations.
$\mathcal{M}$	A family of equation sets.
$\mathcal{M}_{MO}$	A family of MO sets.
$\mathcal{M}_{MSO}$	A family of MSO sets.
ω_m	The set of all feasible minimal rejectable models in a diagnosis model.
$\omega_m(\boldsymbol{z}_0)$	The set of all feasible minimal rejectable models in a diagnosis model and at $\boldsymbol{z}_0$.
C_b	A checking model of a behavioral mode b.
M^*	The proper overdetermined (PO) part of a set M of linear equations.
$G(E, X)$	The bipartite graph with an equation set E and a variable set X as node sets representing the structure of E.
$\mathrm{var}_X(E)$	The variables in X contained in some equation in E.
ν(G)	The size of a maximal matching in a bipartite graph G.
M^+	The proper structurally overdetermined (PSO) part of a set M of equations.
$\bar{\varphi}_s\,(M)$	The surplus of a model M.
$\varphi\,(M)$	The redundancy of a linear model M.
$\varphi_s\,(M)$	The structural redundancy of a model M.
$\mathcal{O}(M)$	The observation set of a model M.

Linear Space and Matrix Notation

rank (A)	The normal-rank of a polynomial matrix A.
s-rank (A)	The structural rank of a matrix A.
$A[I]$	The sub-matrix of a matrix A containing the rows I.
$A[I, J]$	The sub-matrix of a matrix A defined the rows I and the columns J.
$A[:, J]$	The sub-matrix of a matrix A containing the columns J.
N_A	A matrix such that the rows of N_A is an irreducible basis for the left null-space of a polynomial matrix A.
$N_{A[I]}$	A zero-padded matrix such that $N_{A[I]}A = 0$.
$\dim(\mathscr{A})$	The dimension of a space $\mathscr{A}$.
$\mathscr{A}^{\perp}$.	The orthogonal complement to a linear space $\mathscr{A}$.
$\mathrm{sp}(A)$	The row-span of a matrix A.
$\mathrm{Im}(A)$	The column-span of a matrix A.

Diagnosis System Notation

Δ	A diagnosis system represented as a set of tests.
$C(z)$	The set of candidates computed by a diagnosis system with z as input.
δ	A diagnosis test.
$\mathcal{R}$	A subset of $\mathbb{R}$ that defines the rejection region of a diagnosis test.
$\mathcal{O}(\delta)$	The acceptance set of a test δ.
$T(z)$	A test quantity of a diagnosis test.
Φ	A set of system behavioral modes such that sys $\in \Phi$ is the null hypothesis of a test.

Isolability Relation Notation

I_d	A desired isolability.
$I(\Delta)$	The analytical isolability of a diagnosis system Δ.
$I(\mathbb{M})$	The analytical isolability of a diagnosis model $\mathbb{M}$.
$I_p(\langle C_{b_i} \rangle)$	An isolability prediction based on a list of checking models C_{b_i}.
$I_s(\Delta)$	The structural isolability of a diagnosis system Δ.
$I_s(\omega)$	The structural isolability of a set of models ω.
$P(\Delta)$	The analytical candidate implication partial order of Δ.
$P_s(\Delta)$	The structural candidate implication partial order of Δ.

Miscellaneous Notation

$\mathcal{P}(M)$	The power set of a set M.
$\deg(b(p))$	The degree of a polynomial $b(p)$.
$\operatorname{sgn} \pi$	The signature of the permutation π.
$\mathcal{D}'$	The set of distributions.
$\mathbb{R}$,$\mathbb{C}$	The field of real/complex numbers.
$\mathbb{Z}$	The set of integer numbers.
$\mathbb{Z}_+$	The set of positive integer numbers.
$\mathbb{N}$	The set of natural numbers.

1

I

Our modern society depends strongly on reliable complex technological processes. Human safety, environmental, and process protection requirements are some examples of demands that must be fulfilled apart from fulfilling process performance requirements. To meet all these demands, it is important that all parts of a process are functioning correctly according to their design purposes. A *fault* is something that changes the behavior of some part of the process such that this part does no longer fulfill its purpose (Blanke et al., 2003). In complex and automated processes the effects of a fault can quickly propagate and lead to degradation of process performance or even worse to a catastrophic failure. Therefore faults have to be found as quickly as possible and decisions have to be made to avoid process failure by stopping the propagation of their effects and to minimize process performance degradation. To make correct decisions, it is not sufficient to know that a fault has occurred, it is also necessary to know which type of fault that has occurred. To decide whether or not a fault has occurred is called *fault detection* and to determine the type and location of the fault is called *fault isolation*.

The field of diagnosis includes methods for detecting and isolating faults and a device for this purpose is called a *diagnosis system*. A general setup of a diagnosis application is shown in Figure 1.1 with a diagnosis system diagnosing a process. The process, i.e., the system to be diagnosed, is assumed to be working in exactly one of a set of pre-defined modes, here called *system behavioral modes*. The set of pre-defined system behavioral modes includes typically a no-fault mode and some fault modes. The input to the diagnosis system is all available knowledge about the present behavior of the process and this is called an *observation*. An observation consists typically of sensor measurements and controller outputs. The purpose of a diagnosis system is, given observations, to detect and isolate faults in the process.

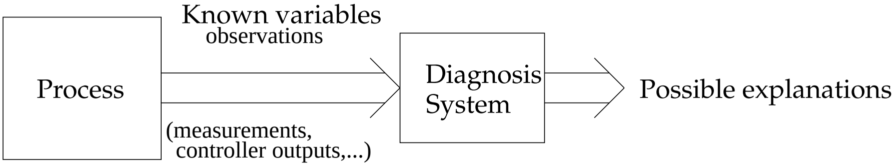

Figure 1.1: A general setup of a diagnosis application with a diagnosis system diagnosing a process.

1.1 Basic Principles of Fault Detection and Isolation

The basic idea of detecting faults is to conclude that an observed behavior is different from the expected behavior of the process when working in the fault-free mode. If this can be concluded, it means that a fault must have occurred. The basic idea of isolating a fault is to conclude that other faults can not produce the observed behavior. By excluding faults as possible explanations, the number of possible faults can be reduced and fault isolation is obtained. By using these principles, the expected behavior of the process when working in different behavioral modes need to be known in order to achieve fault detection and isolation.

The expected behavior of the process can be described in many different ways, but in this thesis we will assume that a mathematical model of the process is used to describe the expected behavior. This is usually refered to as *model-based diagnosis*. We will consider the models to be deterministic in this thesis and typically differential-algebraic systems. To be able to distinguish the behaviors of the process when being in different behavioral modes, it is important to describe both the behavior of fault free operation and how the different faults influence the behavior. A model used to describe the behavior of the process when being in a specific behavioral mode will be called a *behavioral model*. These models are collected in a model called the *diagnosis model* that describes the behaviors of the process for all different behavioral modes.

Since the expected behaviors are described by models, it follows that a fault is detected if the observed behavior is inconsistent with the behavioral model describing the fault free behavior. Isolation is obtained by concluding that that the observed behavior and behavioral models describing different fault modes are inconsistent. In conclusion, both fault detection and fault isolation are obtained by testing if different models are consistent with an observation and this is the principle of *consistency-based diagnosis*.

1.2 Basic Principles for Consistency Checking

To test if a model and an observation are consistent, is to decide if the model can be satisfied given the observation. To be more precise, if z_0 is the observation, x is a vector of unknown trajectories, and $f(z,x) = 0$ is the differential-algebraic model, a mathematical formulation of this decision problem is to determine if there exists an x such that

$$f(z_0, x) = 0 \tag{1.1}$$

This problem is in general difficult to solve. Furthermore, to detect and isolate faults as fast as possible, the time elapsed between getting an observation and determining the consistency of (1.1) must be short. Therefore, it is crucial for fast detection and isolation that the computational complexity of testing the consistency of each model is small.

One way to reduce the computational complexity of testing consistency is to use *residual generators* together with thresholds (Patton et al., 1989; Gertler, 1998). More generally and in accordance with statistical hypothesis testing (Casella and L.Berger, 1990), we use the notion *tests*. A test contains a *test quantity* $T(z_0)$ that is an explicit function that maps observations z_0 into real numbers and a *rejection region* $\mathcal{R} \subset \mathbb{R}$ that is a proper subset of all real numbers (Casella and L.Berger, 1990). The idea is to select these such that

$$T(z_0(t)) \in \mathcal{R} \tag{1.2}$$

implies that (1.1) and z_0 are inconsistent. Since the test quantity is an explicit function, it does not include any unknowns, and it follows that the problem of determining if (1.1) and z_0 is inconsistent is reduced to the procedure of inserting the observed values in the test quantity, compute the value of the test quantity, and compare the computed value with the rejection region. A test makes a binary decision, if the test quantity belongs to the rejection region, then it is concluded that the tested model is inconsistent with the observation and otherwise no conclusion is drawn.

1.3 Architecture of a Diagnosis System

By testing different models describing the behavior of different behavioral modes, fault isolation can be achieved. One naive approach is to use one pre-compiled test for each behavioral model. The architecture of a diagnosis system using this approach is shown in Figure 1.2. In this case, each test decides if the behavioral mode corresponding to the test is a possible explanation or not. The decision from all tests are inputs to the unit "Fault Isolation" in Figure 1.2 that computes the set of possible explanations, i.e., all behavioral modes that can explain the test results.

There are two main disadvantages with testing each behavioral model separately. The first disadvantage is that the number of behavioral modes can be large, especially when considering multiple faults, and then the number of tests will be large. The second disadvantage is that each system behavioral mode specifies the behavior of all parts of the process and this means that each test has to consider a model of the complete system including all sensor and controller signals. Therefore, it can be expected that the computational complexity of running these tests are high.

Both these disadvantages can be handled by testing models that are subsets of equations in behavioral models. In this way, each test uses only the observed behavior of a part of the process, i.e., only some sensor and controller signals are needed as inputs to each test. Furthermore by testing a small part of a process, all system behavioral modes that specify the same expected behavior for this part is tested using this single test. Hence, both the computational complexity for each test and the number of test might be reduced by testing small subsets of equations.

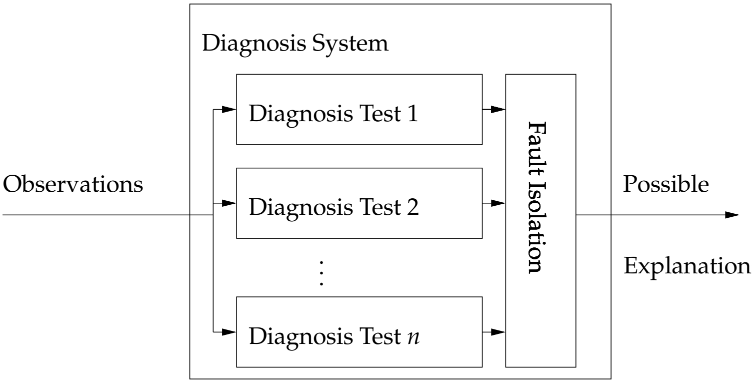

Figure 1.2: Architecture of a diagnosis system.

1.3.1 Good Fault Isolation Capability

When designing a diagnosis system, there is a trade-off between low on-line computational complexity and good fault isolation capability. Here, we will assume that a diagnosis system designer specifies a fault isolability goal and given this goal, the on-line computational complexity is then minimized. In this thesis the designer can express the fault isolation goal by using one of the following two alternative types of fault isolation goals.

A diagnosis model describes all knowledge about the expected behaviors of the process and the first goal is to construct a diagnosis system that uses all this information. By using all knowledge about the expected behaviors of the process, the best possible diagnosis system is obtained. Such diagnosis system will be called a *sound and complete diagnosis system.*

The second goal is to find a diagnosis system with *maximum isolability*, which is a diagnosis system that, for some observation, can distinguish one behavioral mode from another behavioral mode if the diagnosis model supports this.

The first goal is more ambitious than the second goal. For complex systems, the first goal can be too ambitious and then it is possible to reduce the first goal such that all knowledge about the expected behaviors are used only for a subset of behavioral modes. Furthermore, a reduced version of the second goal is to specify exactly which modes that we wish to distinguish from other modes.

1.4 Diagnosis System Construction

The construction of a diagnosis system with the proposed architecture can be divided into the following three main steps:

a) Select models to test such that a chosen fault isolation goal might be fulfilled.

b) Construct a test quantity and a rejection region for each selected model.

c) Design a fault isolation unit.

Step (c) is not the focus of this thesis. For more details about this step see e.g. (Nyberg, 2006; Cordier et al., 2004). In contrast to many previous works within the field on fault diagnosis, the focus of this thesis is on the first of these three steps (Blanke et al., 2003; Chen and Patton, 1999; Patton et al., 2000; Korbicz et al., 2004). However, to understand the basic principles of how to select models in step (a), we first need to go into some details of step (b).

1.4.1 Successful Test Construction

Assume that a model has been selected in step (a) and that a test based on this model is to be constructed according to step (b). The purpose of the test is, as said before, to detect inconsistencies between the model and different observations. This is only possible if *redundancy* is contained in the model, i.e., basically that, there exists a test quantity $T(z_0)$ and a non-empty rejection region $\mathcal{R}$ such that (1.2) is false if the model is consistent with the observation z_0. A model with redundancy typically contains more equations than unknowns such that all unknowns can be eliminated.

It is not sufficient to know the existence of a test quantity for successful completion of step (b), it must also be possible to derive a test quantity with any available method, for example by using analytical redundancy relation based methods (Chow and Willsky, 1984; Frisk and Nyberg, 1999; Staroswiecki and Comtet-Varga, 2001) or observer based methods (Frank, 1994; Kinneart, 1999; Yu and Shields, 1997; Nikoukhah, 1998). For non-linear dynamic models, the construction of a test quantity involves, i) to chose for the considered model a method that is suitable for test construction, ii) to apply the method and construct a test quantity, and iii) to validate the test against simulated and measured signals. These parts often involve both manual work and experiments using data from the real processes. Thus by selecting few models properly in step (a), it is not only possible to reduce the on-line computational complexity of the resulting diagnosis system, it is also possible to save valuable engineering time focusing only on the needed tests.

1.4.2 Objectives of Model Selection

The selection of models in step (a) will be based on the assumption that the tests for the selected models can be designed. If this assumption turns out to be false for some model, the selection step might need to be reconsidered, i.e., we need to iterate between the two steps (a) and (b). It is important to reduce the number of iterations between these step, since this corresponds to time spent on constructing tests that could not be completed. Furthermore, the models selected in each iteration can be different from the models selected in the previous iteration and if the objective is to minimize the number of tests, then it might turn out that tests designed for models selected in previous iterations are not needed. Thus, it is important to reduce the number iterations.

In conclusion, the models should be selected in step (a) such that the following objectives are met:

I) The selected models should contain redundancy such that tests can be derived.

II) The models should be selected such that the resulting tests fulfill the chosen fault isolability goal.

III) The models should be selected such that the on-line computational complexity of the diagnosis system is minimized.

A heuristic for the third objective is to divide this objective into two parts. First, the number of selected models should be minimized. Second, the models should be selected such that each test have low on-line computational complexity. In Section 1.3, it was argued that both the number of tests and the computational complexity for each test might be reduced by testing small models. This and objective (I) implies that the minimal sets of equations that contain redundancy should be especially attractive to select in order to fulfill (I) and (III). These models will be called *minimal rejectable models*.

1.4.3 Finding Model Redundancy Using Structural Methods

When considering nonlinear diagnosis models, it can be a major task to identify models with redundancy and especially the minimal rejectable models. This task is closely related to the problem of variable elimination of the unknowns. For differential-algebraic models, differential gröbner basis (Mansfield, 1991) and characteristic sets (Ritt, 1950) techniques provide automatic elimination methods (Wittkopf, 2004; Mishra, 1993). These algorithms will theoretically terminate with the desired output, but often for individual problems the computations either take an unreasonable time to complete, or require more memory than is available (Wittkopf and Reid, 2001). Furthermore, not all non-linear models that we consider in this thesis are differential polynomials, e.g. the models can also contain for example look-up tables.

To handle non-polynomial differential algebraic models and to cope with the computational complexity of identifying model redundancy in polynomial differential algebraic models, *structural analysis* will be used to investigate model redundancy by means efficient graph-based tools.

The structure of a model contains the information about which variables that are included in each equation, and numerical values and analytical expressions are ignored. Systematic structural approaches to find models with redundancy have been suggested in e.g. (Blanke et al., 2003), (Cassar and Staroswiecki, 1997), (Pulido and Alonso, 2002) (Travé-Massuyès et al., 2001), and (Krysander and Nyberg, 2002a). All these approaches have in common that models with redundancy are found among the models with more equations than unknowns. Furthermore, of all these models, it is the minimal ones that have been used to derive test quantities.

1.5 Main Objective

The main objective of this thesis is to investigate how to systematically and automatically select models that fulfills the three objectives (I)-(III) using structural methods. By solving this problem, a diagnosis system designer can specify a fault isolability capability goal and follow the procedure in Section 1.4 to obtain a diagnosis system with the minimum number of tests needed to fulfill the goal. By selecting models such that objective (I) is fulfilled, no iterations between step (a) and step (b) are needed. This saves valuable engineering time and focuses the test construction work on important parts of the diagnosis model.

1.6 Summary and Contribution of the Thesis

This section summarizes the scope and the organization of the thesis. In Chapter 2, a framework for diagnosis is introduced. The Chapter 3 to 9 are devoted to different aspects of finding models such that a sound and complete diagnosis system can be constructed based on these models. In Chapter 3 to 5, basic ideas are described considering linear static models. In Chapter 6, we also consider linear dynamic systems. The linear investigations are used to establish a formal link between structural methods and linear methods in Chapter 8 and 9. Chapter 7 and 8 shows that structural methods can be applied also for non-linear models. Chapter 10 investigates how to prioritize the selection of dynamic models using structural analysis of dynamic model properties. Finally, the purpose of Chapter 11-12 is to describe how a diagnosis system with maximum isolability can be constructed.

Chapter 2: A Framework for Model Based Diagnosis

A novel framework for model-based diagnosis is proposed using ideas from artificial intelligence (AI) (Hamscher et al., 1992), fault detection and isolation (FDI) (Patton et al., 1989, 2000; Blanke et al., 2003), and statistical hypothesis testing (Casella and L.Berger, 1990). A diagnosis model contains all behavioral models and an equation can be included in several different behavioral models. To avoid multiple copies of this equation in the diagnosis model, we include the information of which behavioral models that the equation is included in, i.e., how the validity of the model equation depends on the behavioral modes.

The tests are, as said before, assumed to be computed off-line as in FDI (Patton et al., 1989; Gertler, 1998). It is shown how standard FDI methods, such as residuals based on parity relations (Chow and Willsky, 1984; Frisk and Nyberg, 1999; Staroswiecki and Comtet-Varga, 2001) or observer based approach (Frank, 1994; Kinneart, 1999; Yu and Shields, 1997; Nikoukhah, 1998), can be used within the framework for test construction. The type of tests that are used are standard hypothesis tests from statistical hypothesis testing theory and it is possible to treat noise in a sound way. That is, even in a noisy system, faults can be correctly isolated.

In Section 1.4, diagnosis system construction using tests based on different small models was outlined and in Chapter 2 further details are discussed. We introduce, the two important properties of a diagnosis system related to the first goal to obtain good fault isolation capability of a diagnosis system, i.e., complete and sound. A diagnosis system is *complete* if all possible explanations are contained in the output from the diagnosis system. Contrary, the diagnosis system is *sound* if only possible explanations are contained in the output.

We present guidelines of how to construct each individual test in a diagnosis system such that the resulting diagnosis system becomes complete. A sound and complete diagnosis system exactly computes the set of all possible explanations and this implies that any inconsistency in any part of the diagnosis model must be detected. For such diagnosis system, it is required that there are tests that check the consistency of every redundant part of the diagnosis model. A key result is a sufficient and necessary condition for which set of models that tests can be based on, such that a sound and complete diagnosis system exists. This condition will later be referred to as the soundness-criteria. This soundness-criteria can be used to compute the minimum number of tests in a sound and complete diagnosis system, and also to find which models to test. It is shown that the number of tests in general can be decreased by checking the consistency of small models, and then especially the minimal sets of equations that contain redundancy. These models will be called minimal rejectable models. In Chapter 3, 6, and 7 the solutions including minimal rejectable models to the soundness-condition for linear static, for linear dynamic, and for general dynamic models are investigated respectively.

Chapter 3: Soundness when Diagnosing Linear Static Systems

This chapter presents a new theory and algorithms for finding the minimum set of models that is needed to be tested in order to obtain a sound and complete diagnosis system when considering linear static diagnosis models. This is done by investigating which sub-models that can be tested to obtain soundness. A key contribution is that there exists a sound and complete diagnosis system based on the set of all minimal rejectable models in a diagnosis model. In general, not all minimal rejectable models need to be tested to obtain soundness, and an algorithm for finding all minimal sufficient subsets of minimal rejectable models is proposed.

Chapter 4: An Efficient Algorithm for Finding all MO Sets

This chapter presents an new efficient algorithm for computing all minimal rejectable models in a linear static or dynamic model. The input to this algorithm is a set of equations that contain redundancy. The algorithm is based on a top-down approach in the sense that we start with all equations and then remove equations step by step until a minimal rejectable model is found. The algorithm is constructed such that the combination of equations that are removed are exactly those combinations that need to be removed to find each minimal rejectable model once and only once.

Chapter 5: An Algorithm for Finding all Feasible MO Sets

In a general diagnosis model there can be set of equations with inconsistent validity. For example, some equation is valid only in the no-fault mode and another only in some fault mode. A model with inconsistent validity, do not describe the behavior in any behavioral mode and is not useful for diagnosis. Thus, models with consistent validity will be considered and these models will here be called *feasible* models.

The algorithm presented in Chapter 4, does not consider the validity of equations. This means that if this algorithm is applied to a general diagnosis model, minimal rejectable models with inconsistent validity are found if such models exist.

This chapter presents a novel algorithm that handles the validity of the equations such that only models with consistent validity are found. This is done by restricting which equations that are allowed to be removed when applying the algorithm presented in Chapter 4. In this way, the property that all minimal rejectable models are found once and only once is transfered to the extended algorithm. The algorithm presented in this chapter is not limited to linear models if an algorithm handling also non-linear models is used instead of the algorithm in Chapter 4.

Chapter 6: Soundness when Diagnosing Linear Dynamic Systems

This chapter presents new theory and algorithms for linear dynamic models corresponding to the presentation given in Chapter 3 for linear static models. A key result is that there exists a sound and complete diagnosis system where the tests are based on the set of all minimal rejectable models with consistent validity. All these minimal rejectable models can be found by the algorithm presented in Chapter 5. It will also be shown that it is in general not necessary to test all minimal rejectable models with consistent validity. Theory for selecting and an algorithm for finding a smallest subset of minimal rejectable models are therefore developed.

It is also shown under a mild rank condition on the diagnosis model that given a minimal rejectable model, the behavioral modes that influence any test quantity derived from the model are given by the validity of the equations. Hence no further fault influence analysis of each test quantity is needed.

Chapter 7: Soundness when Diagnosing General Systems

This chapter presents solutions to the soundness-criteria when considering a diagnosis model with general non-linear equations. Contrary to the linear chapters, methods for finding rejectable models are not proposed. Here, it is assumed that it is possible to find all feasible minimal rejectable models in the non-linear model by using some existing technique. Even though it might be difficult to compute all minimal rejectable models, we show that there exists a sound and complete diagnosis system with tests based on the set of all minimal rejectable models with consistent validity. Furthermore a sufficient and necessary condition for which set of models that is sufficient to test is given. An example shows how this result can be used to minimize and select models such that a sound diagnosis system can be obtained with the minimum number of tests. Furthermore, an algorithm is proposed that given all minimal rejectable models finds all minimal solutions to the soundness-criteria.

Chapter 8: Finding Rejectable Models Using Structural Methods

Systematic structural approaches to find models with redundancy have been suggested in e.g. (Blanke et al., 2003), (Cassar and Staroswiecki, 1997), (Pulido and Alonso, 2002) (Travé-Massuyès et al., 2001), and (Krysander and Nyberg, 2002a). All these approaches have in common that models with redundancy are found among the models with more equations than unknowns. Furthermore, of all these models, it is the minimal ones that have been used to derive test quantities. Such models that also contains known variables will here be called *minimal structurally overdetermined (MSO) sets* of equations.

In this chapter, we will formally show for linear systems that a model is an MSO sets if and only if the model is a minimal rejectable model in the generic case. Three different structural representations of dynamic models are recalled from the literature. It is exemplified that MSO sets correspond to minimal rejectable models for all three representations. The difference between the two main representations is whether different order of derivatives of the same signal should be considered to be different independent signals or not. The structural model properties are applicable also to non-linear dynamic models, and interpreted in a similar way as for linear systems. We give examples where the MSO sets are the minimal models that analytical redundancy relations can be derived from by using elimination tools, i.e., the idea of testing MSO sets can be extended to the non-linear dynamic case.

Chapter 9: An Efficient Algorithm for Finding all MSO Sets

A main contribution in this thesis is a new efficient algorithm for computing all MSO sets in a model. The proposed algorithm is similar to the algorithm for finding all minimal rejectable models described in Chapter 4. The only difference is that rank operations are replaced by corresponding graph theoretical operations. This algorithm can be used in combination with the algorithm presented in Chapter 5, and then all MSO sets with consistent validity are found.

The proposed algorithm can use any structural representation of dynamic systems that are recalled in Chapter 8 for finding models with redundancy. For complexity comparison, previous algorithms are recalled. Contrary to all previous algorithms, this algorithm uses a top-down approach and it is shown that the time complexity under certain conditions is much better for the new algorithm. This is illustrated using a Scania truck engine model.

Chapter 10: Structural Analysis of MSO Sets of Differential-Algebraic Equations

When finding MSO sets in differential-algebraic systems the structural representations that do not distinguish between different order of derivatives is the most compact representation. To find all MSO sets in this representation is therefore computationally less demanding than using the other expanded structural representation where different order of derivatives are considered to be separate independent variables. However, the expanded representation provides more information about the differential algebraic system.

In this chapter, we consider an MSO set in the original representation and shows how an MSO set in the expanded structural representation can be obtained. The main reason for doing this is that the corresponding differential-algebraic model is transformed into an algebraic model. Then test quantities of the differential algebraic model can be obtained also be using static methods applied to the corresponding algebraic model.

The extended MSO set is obtained by differentiating the equations included in the original MSO set. It is desirable to differentiate the equations as few times as possible, to avoid higher derivatives of measured signals and look-up tables. A key result is that there exists a unique expanded MSO set where all equations are differentiated less number of times than in any other expanded MSO set. An algorithm is presented that given an MSO set in the original representation returns this unique expanded MSO set. This algorithm is purely structural and is based on the new concept of *structural differentiation*.

Chapter 11: Fault Isolability Analysis of Diagnosis Systems

In this and the following chapter the focus will be on the second goal to achieve good fault isolation capability, i.e., to find a diagnosis system with maximum isolability. A design procedure for test selection based on isolability properties is proposed. Two type of isolability goals can be specified. Either the maximum isolability is the goal or a desired isolability specified by the diagnosis system designer is the goal. The designer has to provide a set of potential test and by following the procedure a minimum subset the potential tests is selected with the desired isolability or the best possible isolability. It is shown that the test quantities and the rejection regions of the potential tests do not need to be constructed in order to decide that the tests are not needed. This means that test quantities and rejection regions have to be derived only for the selected tests. By computing which faults that influence each selected test, the procedure determines if the desired or the best possible isolability has been obtained.

Chapter 12: Fault Isolability Prediction of Diagnosis Models and its Applications

In the development of processes including diagnosis, design decisions are taken, e.g. sensor configuration selection, which affects the fault isolability possibilities. In this chapter an algorithm for predicting fault isolability possibilities using a structural model describing the process is proposed. Since only a structural model is needed as input, the algorithm can easily predict fault isolability possibilities of different design concepts. In contrast to previous algorithms using structural models no assumption is imposed on the model. The algorithm computes faults that cannot be distinguished from other faults, which can be used to exclude design alternatives with insufficient isolability possibility.

Furthermore, a design procedure for constructing a diagnosis system with the maximum possible fault isolation capability is given. This design procedure is based on isolability predictions in combination with the theory for selecting tests in Chapter 11. The fault isolability of a diagnosis system with maximum possible fault isolation capability and the diagnosis model is exactly the same. Hence the fault isolation possibilities of a diagnosis model can be computed in this way.

1.7 The Results from a User Perspective

In this section, we will give an overall picture of how the design of diagnosis systems can be done and also point out where results from this thesis can be applied. To encourage the use of the results, it is highly desirable to have software support in the design procedure. The ongoing work for determining the architecture for such a toolbox is described in (Frisk et al., 2006).

The procedure for designing a diagnosis systems contains several steps. Important steps are the following:

a) Deriving fault isolability requirements.

b) Constructing a diagnosis model of the process.

c) Performing detectability and isolability analysis on the diagnosis model.

d) Selecting models for test construction.

  - Compute minimal rejectable models.
  - Specify if soundness or maximum isolability should be the design goal.
  - Select a minimum cardinality set of models that may fulfill the chosen goal.

e) Designing tests for each selected model.

  - Design a test quantity with an appropriate method.
  - Compute a proper rejection region by using simulations and/or measurements.
  - Evaluate the fault influence of the test and if needed go back to step (d).

f) Selecting fault isolation algorithm.

Next the different steps will be described. The focus of this thesis is on steps (c) and (d) and the descriptions of these steps are therefore more detailed than the other steps.

1.7.1 Deriving Fault Isolability Requirements

In step (a), different faults of the process are identified. This can be done by using fault tree analysis that is a systematic way to investigate credible causes for an undesired event in a process (Stamatelatos and Vesley, 2002; Vesley et al., 1981) or by using failure mode and effect analysis (Stamatis, 1995). By evaluating how the different faults effects the process behavior, intolerable consequences is found and fault isolability requirements can then be specified. The positive effects of including diagnosis can then be evaluated by an extended fault tree analysis according to (Åslund et al., 2006).

1.7.2 Constructing a Diagnosis Model of the Process

In step (b) a diagnosis models is constructed according to the modeling framework that is proposed in Chapter 2. The modeling guidelines proposed in Section 12.5 can also be consider to improve model selection using structural analysis.

1.7.3 Performing Detectability and Isolability Analysis

The purpose of step (c) is to investigate if there can exist some diagnosis system with the required fault isolability. This analysis can be done with the methods proposed in Chapter 12. If the answer is no, additional fault modeling or additional sensors must be used and this is exemplified in Chapter 12. If the fault isolability requirements can be met, then diagnosis system construction can begin.

1.7.4 Selecting Models for Test Construction

The first part in step (d) is to compute all minimal rejectable models. If the diagnosis model is linear, then linear methods described in Chapter 3-6 can be used. If the diagnosis model is non-linear, we need to decide if we want to use a linearization of the model and apply the linear methods or if we want to use the structural methods presented in Chapter 8 and 9.

The second part of step (d) is to chose which goals for obtaining good isolability that we will used, i.e., to aim for completeness and soundness or to aim for obtaining maximum isolability.

The third part of step (d) is to select a minimum cardinality subset of minimal rejectable models that may fulfills the chosen goal. If soundness is the goal, then methods in chapters 3, 6, and 7, can be used for linear static, linear dynamic, and non-linear models respectively. If maximum isolability is the goal, the methods in chapters 11, and 12 can be used.

There might exist several possible minimum cardinality sets of minimal rejectable models and this rises the question which set of models to use. In Section 1.4.1, we argued that the models should be selected such that a test can be derived from each model with some existing tool. By using the graph-theoretical algorithm presented in Chapter 10, it is possible to prioritize between dynamic models by predicting dynamical properties and this prioritization can be used for model selection.

1.7.5 Designing Tests for the Selected Models

In step (e), a test quantity and a rejection region are designed for each selected model. Each model needs to be considered separately to use an appropriate tool for test construction. For example, if the model is linear, then linear residual generation can be used to obtain a test quantity (Chow and Willsky, 1984; Nyberg and Frisk, 2006). If the model is polynomial, gröbner basis elimination techniques can be used (Frisk, 2001; Staroswiecki and Comtet-Varga, 2001). If high order of derivatives are included in an analytical redundancy relation, then an observer based method can be tried (Frank, 1994; Kinneart, 1999; Yu and Shields, 1997; Nikoukhah, 1998). Test quantities are derived from the deterministic diagnosis model that might contain model uncertainties, for example that noise has been neglected. Simulations and measurements from a real process can be used to verify the test quantity is applicable to real process data and also be used to select a proper rejection region. When a rejection region has been determined, the set of faults that influence the test can be investigated by implementing different faults in models and simulate the test response. This can also be done by implementing faults on a running process and use the measurement data in the same way. If the fault influence is different from the expected fault influence used in step (c) for model selection, it might be necessary to return to step (c) and make a new selection.

1.7.6 Selecting Fault Isolation Algorithm

Finally, in step (f) a suitable fault isolation algorithm should be selected, i.e., to design the unit "Fault Isolation" in Figure 1.2. If only single faults or few behavioral modes are considered, *structured hypothesis tests* can be used (Nyberg, 1999). If multiple faults with only two modes per component are considered, then the algorithm in (Kleer and Williams, 1987) can be used. If multiple faults and fault modes are considered the isolation algorithm in (Nyberg, 2006) can be used.

By combining the designed tests and the selected isolation algorithm as seen in Figure 1.2, a diagnosis system is obtained.

1.8 Main Contributions

The main contributions are summarized in the following list.

- The framework combining methods from AI and FDI for diagnosis presented in Chapter 2.

- Theorem 7.3 that gives a necessary and sufficient condition expressed in terms of minimal rejectable models for which sets of models that a sound and complete diagnosis system can be based on.

- Corollary 7.1 that states that there exists a sound and complete diagnosis system based on all feasible minimal rejectable models in a diagnosis model.

- Algorithm 4 for finding all minimal rejectable models in a linear static or dynamic diagnosis model.

- Algorithm 7 for finding all feasible minimal rejectable models or all feasible MSO sets.

- Theorem 8.6 that establish a formal link between structural and analytical methods.

- Algorithm 11 for finding all MSO sets in model.

- Algorithms for finding all minimal set of models the fulfills the soundness-criteria. For linear static models Algorithm 2 in combination with Algorithm 1 can be used, for linear dynamic models and Algorithm 2 in combination with Algorithm 8 are applicable, and for general dynamic models Algorithm 9 should be used.

- Algorithm 12 that transforms an MSO set of differential-algebraic equations into an MSO set of algebraic equations.

- The isolability analysis of a diagnosis model using the two structural methods proposed in Section 12.2 and in Section 12.3.

- The procedure in Section 12.7.4 for designing a diagnosis system with maximum isolability and with the minimum number of tests.

1.9 Publications

In the research work, leading to this thesis, the author has produced a licentiate thesis and the following conference and journal papers:

- M. Krysander and M. Nyberg (2002). Structural Analysis utilizing MSS Sets with Application to a Paper Plant, *Proc. of the Thirteenth International Workshop on Principles of Diagnosis*, Semmering, Austria.
- M. Krysander and M. Nyberg (2002). Structural Analysis for Fault Diagnosis of DAE Systems Utilizing MSS Sets, *IFAC World Congress*, Barcelona, Spain.
- M. Krysander and M. Nyberg (2002). Fault Diagnosis utilizing Structural Analysis, *CCSSE*, Norrköping, Sweden.
- M. Nyberg and M. Krysander (2003). Combining AI, FDI, and statistical hypothesis-testing in a framework for diagnosis, *Proceedings of IFAC Safeprocess'03*, Washington, USA.
- E. Frisk, D. Düştegör, M. Krysander, and V. Cocquempot (2003). Improving fault isolability properties by structural analysis of faulty behavior models: application to the DAMADICS benchmark problem. *Proceedings of IFAC Safeprocess'03*, Washington, USA.
- D. Düştegör, E. Frisk, V. Cocquempot, M. Krysander, M. Staroswiecki (2006). Structural Analysis of Fault Isolability in the DAMADICS benchmark. *Control Engineering Practice*, 14(6):597–608.
- M. Krysander, J. Åslund, and M. Nyberg (2005). An Efficient Algorithm for Finding Over-constrained Sub-systems for Construction of Diagnostic Tests. 16th International Workshop on Principles of Diagnosis (DX-05). Pacific Grove, California, USA.
- M. Krysander and M. Nyberg (2005). Fault Isolability Prediction of Diagnostic Models. 16th International Workshop on Principles of Diagnosis (DX-05). Pacific Grove, California, USA.
- M. Krysander, and J. Åslund (2005). Graph Theoretical Methods for Finding Analytical Redundancy Relations in Overdetermined Differential Algebraic Systems. *Proceedings of IMACS World Congress 2005*, Paris, France.
- J. Åslund, J. Biteus, E. Frisk, M. Krysander, and L. Nielsen (2005). A systematic inclusion of diagnosis performance in fault tree analysis. *IFAC World Congress*, Prague, Czech Republic.
- E. Frisk, M. Krysander, M. Nyberg, and J. Åslund (2006). A toolbox for design of diagnosis systems. *Proceedings of IFAC Safeprocess'06*, Beijing, P. R. China.
- J. Åslund, J. Biteus, E. Frisk, M. Krysander, and L. Nielsen. Safety analysis of autonomous systems by extended fault tree analysis. Accepted for publication in: *International Journal of Adaptive Control and Signal Processing*.

2

A F M B D

Fault diagnosis has in the literature been studied from mainly two different perspectives. The first is control theory (here denoted FDI), e.g. see (Gertler and Singer, 1990; Gertler, 1998) and the second is AI, e.g. see (Kleer et al., 1992; Reiter, 1987; Kleer and Williams, 1987; Hamscher et al., 1992). In the field of control theory, the literature on fault diagnosis has mostly been focused on the problem of *residual generation*. That is, given a model of the system, how to off-line construct residual signals that are zero in the fault-free case but sensitive to faults. In the field of AI, the focus has been on fault isolation and how to on-line compute what is here called residuals. In this chapter we show how methods from FDI and AI (or more exactly consistency-based diagnosis) can be combined into a common framework for fault diagnosis that will be used in this thesis. The framework proposed is also based upon ideas from statistical hypothesis testing in accordance with the method structured hypothesis tests from (Nyberg, 2002a, 1999).

The modeling of the system to be diagnosed, and the isolation of faults, follows mainly ideas from AI (Dressler et al., 1993). The key point here is to add information in the model of how the validity of each model equation depends on which faults that are present in different components. Isolation is then performed by propagating this information through the diagnosis system. However, one difference is that residuals are assumed to be computed off-line as in FDI. Therefore the on-line machinery can be made more simple, e.g. there is no need to use a so called ATMS (Assumption based Truth Maintenance System) which is common in AI (Kleer and Williams, 1987). All decisions taken in the diagnosis system are based on the theory of statistical hypothesis testing. This means for example that noise and uncertainties can be handled in a sound way.

By combining these ideas from FDI, AI, and hypothesis testing, we will obtain a framework that is able to efficiently handle: fault models, several different fault types (e.g. parameter- and additive faults), more than two be-

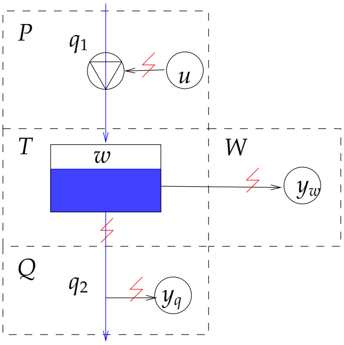

Figure 2.1: The system to be diagnosed. The location of possible faults are denoted with a red flash.

havioral modes per component, general differential-algebraic models, noise, uncertainties, decoupling of disturbances, static and dynamic systems, and isolation of multiple faults.

The modeling framework and how information about different faults is incorporated in the model are described in Section 2.1. The design of a diagnosis system is then presented in Sections 2.2 and 2.3. The connection to FDI methods are more explicitly elaborated in Section 2.4. Finally, Section 2.5 discusses the output from the diagnosis system.

2.1 Modeling Framework

This section describes the modeling framework that is later used in the construction of the diagnosis system. Using this modeling framework, all information about the faults are included in the model. This fault information is then the basis for the reasoning about faults.

Throughout the chapter, we will exemplify some concepts and techniques on the following example.

Example 2.1
The example chosen is shown in Figure 2.1 and represents a water-tank system. This system has a pump that is pumping water into the top of a tank. The pump is controlled by a control signal u. The water flows out of the tank through a pipe connected to the bottom of the tank. The flows into and out of the tank are denoted q_i, and the water-level in the tank is denoted w. The water-level in the tank and the outflow of the tank are measured by sensors. The water-level sensor signal is y_w, and the outflow sensor signal is y_q.

2.1.1 Components

We assume that the system consists of a set **C** of components. The behavior of each component, and the relation to its outer world, are described by a number of relations. A variable in the relations for a component is either an internal variable or an external variable. External variables are variables that

Table 2.1: Possible behavioral modes for each component of the water-tank system.

Component	Possible Behavioral Modes
Pipe	$P \in \{\mathrm{NF}, \mathrm{UF}\}$
Tank	$T \in \{\mathrm{NF}, \mathrm{C}, \mathrm{UF}\}$
Water-level sensor	$W \in \{\mathrm{NF}, \mathrm{UF}\}$
Flow sensor	$Q \in \{\mathrm{NF}, \mathrm{B}, \mathrm{UF}\}$

are shared with connected adjacent components or can be observed. Internal variables are only known within the component itself. Another alternative, not exploited here, is to consider all variables to be internal variables and then describe interconnections between components explicitly by using equations, e.g. the object-oriented modeling-language Modelica.

Example 2.2
The water-tank system introduced in Example 2.1 and shown in Figure 2.1 consists of four components, i.e. a pump, a tank, a water-level sensor, and a flow sensor. These components will be denoted P, T, W, and Q respectively and are illustrated in Figure 2.1 by the four dashed boxes. This means that the set $\mathbf{C}$ of components is $\mathbf{C} = \{P, T, W, Q\}$ in this case.

2.1.2 Component Behavioral Modes

The behavior of a component can vary depending on which *behavioral mode* the component is in. Different type of faults are typically considered to be different behavioral modes. Examples of behavioral modes for a sensor are no-fault, short-cut, bias, and unknown fault. Abbreviations like NF for no-fault, C for clogging, B for bias, and UF for unknown fault will be used. Furthermore, if for example c is a component then c will also with a little abuse of notation be a variable describing which behavioral mode the component is in. For example $c = \mathrm{NF}$ will denote that c is in behavioral mode NF. The set of all possible behavioral modes for component c is denoted by $\mathbf{B}_c$. It is assumed that a component is in exactly one of its possible behavioral modes, i.e. $c \in \mathbf{B}_c$.

Example 2.3
For the water-tank system, the four components are assumed to have the possible behavioral modes shown in Table 2.1.

2.1.3 Component Behavior

The behavior of each component is as said before described using relations. That is, for each component c there is a set of relations $M_c = \{e_{i_c}, e_{i_c+1}, e_{i_c+2}, ...\}$ describing the behavior of that component. The validity of each relation can in some cases depend on which behavioral mode the component is in. For the water-tank example, we can have the case that a relation $y_q = q_2$ holds if component Q is in behavioral mode NF, i.e. $Q = \mathrm{NF}$, but not necessarily if $Q = \mathrm{B}$. This means that together with each relation e_{i_c}, there is an assumption of the type $c = F1$ (or a disjunction $c = F1 \vee c = F2 \vee \ldots$) that must be fulfilled before the relation e_{i_c} can be assumed to hold.

Example 2.4
The assumptions and the relations for all components of the water-tank system are shown in Table 2.2. The equations describe the pump, the conservation of volume in the tank, the outflow from the tank caused by the gravity derived from *Bernoulli's* equation, the outflow from the tank caused by the gravity when a clogging fault is present, a fault model for a constant clogging fault f_c, the water-level measurement, the outflow measurement, the outflow measurement when the sensor has a bias fault and a fault model for the outflow-measurement fault f_{yq}. The first equation describing the behavior of the tank has no assumption, i.e. that equation is always true.

Table 2.2: An analytical model for the water-tank system shown in Figure 2.1.

Assumption	Equation	Expression
Pump		
$P = \text{NF}$	e_1	$u = q_1$
Tank		
	e_2	$\dot{w} = q_1 - q_2$
$T = \text{NF}$	e_3	$w = q_2^2$
$T = \text{C}$	e_4	$w = (1 - f_c)q_2^2$
$T = \text{C}$	e_5	$\dot{f_c} = 0$
Water-level sensor		
$W = \text{NF}$	e_6	$y_w = w$
Flow sensor		
$Q = \text{NF}$	e_7	$y_q = q_2$
$Q = \text{B}$	e_8	$y_q = q_2 + f_{yq}$
$Q = \text{B}$	e_9	$\dot{f}_{yq} = 0$

2.1.4 System and System Behavioral Modes

The set of all available equations for the system is supposed to describe all information about the behavior of the system. The set of all equations is denoted $\boldsymbol{M}$ and is equal to the union of all equations describing the components, i.e.

$$\boldsymbol{M} = \bigcup_{c \in \mathbf{C}} M_c$$

Further on it needs to be defined which variables in $\boldsymbol{M}$ that are possible to observe, i.e. which variables that are known and which variables that are unknown. The set of known variables is denoted $\boldsymbol{Z}$ and the set of unknown variables is denoted $\boldsymbol{X}$. For the model in Table 2.2, $\boldsymbol{Z} = \{u, y_w, y_q\}$ and $\boldsymbol{X} = \{q_1, q_2, w, f_c, f_{yq}\}$.

The models can be static, dynamic time-discrete, or dynamic time-continuous. For each type of model all variables are assumed to have equal domains. For the static case, the domain of each variable is assumed to be all real numbers $\mathbb{R}$.

For the dynamic time-discrete case, the domain of each variable is assumed to be all time-discrete signals for some finite or infinite time-interval where each time-sample is in the range of $\mathbb{R}$. Finally, for the dynamic time-continuous case, the variables are assumed to be time-continuous signals with range $\mathbb{R}$ and for some finite or infinite time-interval.

For any of the three cases, let $\mathbf{x}$ and $\mathbf{z}$ be the vectors of the variables in $\boldsymbol{X}$ and $\boldsymbol{Z}$ respectively. The domains of $\mathbf{x}$ and $\mathbf{z}$ are then denoted $\mathbb{X}$ and $\mathbb{Z}$ respectively. For the model in Table 2.2, $\mathbf{x}$ and $\mathbf{z}$ are time-continuous signals.

As well as defining behavioral modes for components, it is possible to define behavioral modes called *system behavioral-modes* for systems. A system behavioral-mode completely determines the behavioral mode of all components in the system, i.e. $\wedge_{c\in\mathbf{C}}(c = b_c)$ where $b_c \in \mathbf{B}_c$ for each $c \in \mathbf{C}$. This is also called a *complete mode assignment* (Dressler and Struss, 1994).

An example of a system behavioral-mode for the water-tank system is $P = \mathrm{NF} \wedge T = \mathrm{C} \wedge W = \mathrm{NF} \wedge Q = \mathrm{B}$ meaning that P is in behavioral mode NF, T in C, W in NF, and Q in B. If an ordering of the components is assumed, an alternative representation is to write a system behavioral-mode using a tuple, e.g. $\langle P, T, W, Q\rangle = \langle \mathrm{NF}, \mathrm{C}, \mathrm{NF}, \mathrm{B}\rangle$. The notation $sys = \langle \mathrm{NF}, \mathrm{C}, \mathrm{NF}, \mathrm{B}\rangle$ will be used to denote that the system is in system behavioral-mode $\langle \mathrm{NF}, \mathrm{C}, \mathrm{NF}, \mathrm{B}\rangle$. If there is a components with only one component behavioral mode, then the behavioral mode that this component is in, can for notational convenience be omitted in the tuple.

A system behavioral mode is said to be a single fault mode if exactly one component is in a faulty mode and all other modes are in the no-fault mode. In a similar way double faults and multiple faults refer to the number of components that are in a faulty mode. For the example $\langle \mathrm{NF}, \mathrm{C}, \mathrm{NF}, \mathrm{NF}\rangle$ is a single fault mode and $\langle \mathrm{NF}, \mathrm{C}, \mathrm{NF}, \mathrm{B}\rangle$ is a double and a multiple fault mode.

Like component behavioral-modes, we can use abbreviations to denote system behavioral-modes. This is especially practical when only single-faults are considered. For example for the water-tank system, the system behavioral modes $\langle P, T, W, Q\rangle = \langle \mathrm{NF}, \mathrm{NF}, \mathrm{NF}, \mathrm{NF}\rangle$ and $\langle P, T, W, Q\rangle = \langle \mathrm{UF}, \mathrm{NF}, \mathrm{NF}, \mathrm{NF}\rangle$ can be written **NF** and $\mathbf{UF}_P$.

Another case when it is practical to use abbreviations for system behavioral modes is when each behavioral mode only have one no-fault mode and one faulty mode and multiple faults are considered. Then the faulty components denote the corresponding behavioral mode. For example $\langle P, T, W, F\rangle = \langle \mathrm{UF}, \mathrm{NF}, \mathrm{NF}, \mathrm{NF}\rangle$ and $\langle P, T, W, F\rangle = \langle \mathrm{UF}, \mathrm{UF}, \mathrm{NF}, \mathrm{NF}\rangle$ can be written as **P** and **PT** respectively.

The set of all possible system behavioral modes will be denoted by $\mathbf{B}$. For the water-tank example, the set of all system behavioral modes $\mathbf{B}$ is defined to be all $2^2\, 3^2 = 36$ system behavioral-modes $\mathbf{B} = \{\mathrm{NF}, \mathrm{UF}\} \times \{\mathrm{NF}, \mathrm{C}, \mathrm{UF}\} \times \{\mathrm{NF}, \mathrm{UF}\} \times \{\mathrm{NF}, \mathrm{B}, \mathrm{UF}\}$. For the first equation in Table 2.2 the corresponding assumption is $P = \mathrm{NF}$. This assumption defines a set $\{\mathrm{NF}\} \times \{\mathrm{NF}, \mathrm{C}, \mathrm{UF}\} \times \{\mathrm{NF}, \mathrm{UF}\} \times \{\mathrm{NF}, \mathrm{B}, \mathrm{UF}\} \subset \mathbf{B}$ which is the set of system behavioral-modes where $P = \mathrm{NF}$. This set will be denoted $\phi(P = \mathrm{NF})$. To each equation $e \in \boldsymbol{M}$ there will be a corresponding set of system behavioral-modes denoted $\mathrm{assump}(e)$ for which equation e can be assumed to be valid, i.e.

$$\mathrm{assump} : \boldsymbol{M} \mapsto \mathcal{P}(\mathbf{B})$$

where $\mathcal{P}$ denotes the power set. A relation and its corresponding assumption can then be written

$$\text{sys} \in \text{assump}(e) \rightarrow e \tag{2.1}$$

For the first equation e_1 in Table 2.2, $\text{assump}(e_1) = \phi(P = \text{NF})$.

Finally, all introduced sets and the mapping assump are collected into a tuple $\langle \boldsymbol{M}, \boldsymbol{X}, \mathbb{X}, \boldsymbol{Z}, \mathbb{Z}, \mathbf{B}, \text{assump}\rangle$. This tuple is said to be a *diagnosis model* and will be denoted $\mathbb{M}$.

2.1.5 Diagnosis

To be able to diagnose a system, values of the known variables from the system and a diagnosis model of the system are needed. The known variables are typically measured sensor signals and actuator signals. The information about the known signals will in this thesis be of one of the following three different types:

- a snap-shot
- several sampled values in a time-window
- a continues function in a time-window

All these types will be exploit in later chapters. A snap-shot, some sampled values, or a continuous functions recorded for the diagnosis task will be called an *observation*, i.e. a value $\mathbf{z}_0$ of $\mathbf{z}$.

A set $M \subseteq \boldsymbol{M}$ of equations is called a *model*. A model $M \subseteq \boldsymbol{M}$ is said to be *consistent* with an observation $\mathbf{z} \in \mathbb{Z}$ if

$$\exists \mathbf{x} \in \mathbb{X} : \wedge_{e \in M} e(\mathbf{x}, \mathbf{z}) \tag{2.2}$$

For notational convenience, $\wedge_{e \in M} e(\mathbf{x}, \mathbf{z})$ will be abbreviated $M(\mathbf{x}, \mathbf{z})$. Given an observation, a model is said to be *inconsistent* if the observation is not consistent with the model. The set of consistent observations for a model M is defined as

$$\mathcal{O}(M) = \{\mathbf{z} \in \mathbb{Z} | \exists \mathbf{x} \in \mathbb{X} : M(\mathbf{x}, \mathbf{z})\} \tag{2.3}$$

and is called the *observation set* of M. To determine if $\mathbf{z} \in \mathcal{O}(M)$ for a given observation $\mathbf{z}$ will be called to check the consistency of a model M.

Example 2.5
If the equations in Table 2.2 are considered and $M = \{e_3, e_6, e_7\}$, then $\mathbf{z} \in \mathcal{O}(M)$ if and only if $y_w = y_q^2$, i.e. an equation obtained when eliminating all unknown variables in M.

It is assumed that the system behaves in accordance with the diagnosis model. This implies that an especially important type of model is the type that describes the behavior of the system when it is working in a behavioral mode. Given a behavioral mode $b \in \mathbf{B}$, a model

$$M_b = \{e \in \boldsymbol{M} | b \in \text{assump}(e)\} \tag{2.4}$$

is a *behavioral model* for b.

Example 2.6
For the water-tank example in Table 2.2, the behavioral model for **NF** is $M_{\mathbf{NF}} = \{e_1, e_2, e_3, e_6, e_7\}$ and the behavioral model for $\mathbf{UF}_P$ is $M_{\mathbf{UF}_P} = \{e_2, e_3, e_6, e_7\}$.

Behavioral models will next be used to define which behavioral modes that are consistent with an observation.

Definition 2.1 (Diagnosis). *Given an observation* $\mathbf{z} \in \mathbb{Z}$*, a behavioral mode* $b \in \mathbf{B}$ *is a* diagnosis *if its behavioral model* M_b *is consistent with* $\mathbf{z}$*.*

Note that this definition of diagnosis is equivalent to the one used in consistency based diagnosis (Hamscher et al., 1992). The set of behavioral modes that are diagnoses given an observation $\mathbf{z}$ is denoted

$$\mathcal{D}(\mathbf{z}) = \{b \in \mathbf{B} | \mathbf{z} \in \mathcal{O}(M_b)\} \tag{2.5}$$

The set of observations consistent with the diagnosis model, i.e. consistent with at least one of the behavioral models, can be expressed as

$$\cup_{b \in \mathbf{B}} \mathcal{O}(M_b)$$

and is a subset of $\mathbb{Z}$. If it is a proper subset, important behavioral modes can have been neglected. From now on we will assume that equality holds, i.e.

$$\mathbb{Z} = \cup_{b \in \mathbf{B}} \mathcal{O}(M_b) \tag{2.6}$$

This is no restriction since behavioral modes describing unknown faults can be added to obtain (2.6).

2.2 Diagnosis Tests

A *diagnosis system* is assumed to consist of a set of *diagnosis tests* which is a special case of a general statistical *hypothesis test* (Casella and L.Berger, 1990) and a procedure to compute consistent behavioral modes by using the outcome of the tests. This idea has been described as *structured hypothesis tests* (Nyberg, 2002a). We will in this section discuss diagnosis tests and later, in Section 2.3, describe how several diagnosis tests are combined to form a diagnosis system.

To define a diagnosis test we need the notion of a *test quantity* $T_i(\mathbf{z})$ which is a function from the observations $\mathbf{z}$ to a scalar value. A *diagnosis test* for a noise free model can then be defined as follows:

Definition 2.2 (Diagnosis Test, δ_i). *Let* $\Phi_i \subseteq \mathbf{B}$ *and let* sys *denote the system behavioral mode that the system to be diagnosed is in. A* ***diagnosis test*** δ_i *for the null hypothesis* $H_i^0 : \text{sys} \in \Phi_i$ *is a hypothesis test consisting of a test quantity* $T_i : \mathbb{Z} \mapsto \mathbb{R}$ *and a rejection region* $\mathcal{R}_i \subset \mathbb{R}$ *such that*

$$\text{sys} \in \Phi_i \rightarrow T_i(\mathbf{z}) \notin \mathcal{R}_i \tag{2.7}$$

The complement of the null hypothesis is called the alternative hypothesis and is denoted by $H_i^1 : \text{sys} \notin \Phi_i$. Definition 2.2 means that if $T_i(\mathbf{z}) \in \mathcal{R}_i^C$, sys $\in \Phi_i$ can not hold. This is the same thing as saying that the null hypothesis H_i^0 is rejected and the alternative hypothesis H_i^1 is accepted. The statement sys $\in \Phi_i$ becomes in this case a so called conflict (Kleer and Williams, 1987), i.e. an expression in behavioral modes that is in conflict with the observations.

Example 2.7

For the water-tank example, consider a diagnosis test δ_1 for the null hypothesis

$$H_1^0 : (T = \text{NF}) \wedge (W = \text{NF}) \wedge (Q = \text{NF}) \tag{2.8}$$

i.e. $\Phi_1 = \phi(T = \text{NF} \wedge W = \text{NF} \wedge Q = \text{NF})$. According to the model in Table 2.2 we have that sys $\in \Phi_1$ implies that the equations in $\{e_2, e_3, e_6, e_7\}$ are true. From these equations, it is possible to derive that

$$y_w - y_q^2 = 0 \tag{2.9}$$

This means that a test quantity can be defined at time t as

$$T_1(t) = y_w(t) - y_q(t)^2 \tag{2.10}$$

and a rejection region $\mathcal{R}_1 = \{x | x \notin [-0.1\; 0.1]\}$ implies that

$$\text{sys} \in \Phi_1 \rightarrow T_1(t) = y_w(t) - y_q(t)^2 = 0 \notin \mathcal{R}_1 \tag{2.11}$$

That is, these choices of T_1 and $\mathcal{R}_1$ fulfill the criterion (2.7) for being a diagnosis test for the null hypothesis in (2.8). When $|T_1| > 0.1$ we reject the null hypothesis sys $\in \Phi_1$ and draw the conclusion sys $\notin \Phi_1$, or as expressed in logic (Nyberg and Krysander, 2003)

$$\neg(T = \text{NF} \wedge W = \text{NF} \wedge Q = \text{NF}) \simeq T \in \{\text{C}, \text{UF}\} \vee W = \text{UF} \vee Q \in \{\text{B}, \text{UF}\} \tag{2.12}$$

Note that to evaluate $\simeq$, the assumption that each component is in exactly one of its component behavioral modes, i.e. the assumptions for W, Q, and T shown in Table 2.1 must be used.

No conclusion is drawn from a test in which the null hypothesis has not been rejected. That is, to reject null hypotheses is the only way the diagnosis system can draw conclusions. Note that it is usually not true that sys $\in \Phi_i$ holds when H_i^0 : sys $\in \Phi_i$ is not rejected. It would sometimes be possible to assume something else. However, it is in general difficult (or impossible) to construct $T_i(z)$ and $\mathcal{R}_i$ so that such a conclusion can be drawn when the null hypothesis is not rejected.

Another reason why no conclusion is drawn when the null hypothesis is not rejected is that it is not needed. If there is a conclusion that really can be drawn from $T_{i_1}(z) \in \mathcal{R}_{i_1}^C$, it is always possible to add another diagnosis test δ_{i_2} to the diagnosis system such that this conclusion can be drawn anyway. The suggested framework does not allow us to draw a conclusion when a null hypothesis is not rejected, but this desired conclusion can be obtained if we instead add another test δ_{i_2} with $\Phi_{i_2} = \Phi_{i_1}^C$, $T_{i_2} = T_{i_1}$, and $\mathcal{R}_{i_2} = \mathcal{R}_{i_1}^C$.

2.2.1 Diagnosis Tests and the Model

The idea of model-based diagnosis is to utilize the diagnosis model $\mathbb{M}$ in the construction of the diagnosis tests. For each diagnosis test δ_i, not necessarily all equations in $\boldsymbol{M}$ are utilized as seen in Example 2.7. Instead only a subset $M_i \subset \boldsymbol{M}$ might be considered. This means that, in addition to Φ_i, $T_i(\mathbf{z})$, and $\mathcal{R}_i$, also a model M_i is considered when constructing a diagnosis test. Next we will discussed how $T_i(\mathbf{z})$, $\mathcal{R}_i$, Φ_i, and M_i can be related to fulfill the basic requirement (2.7). To do this the operator assump is first generalized.

In Section 2.1.4, the notion assump(e) was used to pick out the system behavioral modes that implies the relation e. Here we will use assump to pick out the assumption also for a set of model relations as follows

$$\text{assump}(M) = \bigcap_{e \in M} \text{assump}(e) \tag{2.13}$$

With the extended assump operator, we are now able to formulate two guidelines for ensuring that the requirement (2.7) is fulfilled.

a) The set M_i and the null hypothesis Φ_i should satisfy

$$\Phi_i \subseteq \text{assump}(M_i) \tag{2.14}$$

or even better,

$$\Phi_i = \text{assump}(M_i) \tag{2.15}$$

b) The set M_i, the test quantity $T_i(\mathbf{z})$, and the rejection region $\mathcal{R}_i$ should satisfy

$$\mathbf{z} \in \mathcal{O}(M_i) \rightarrow T_i(\mathbf{z}) \in \mathcal{R}_i^C \tag{2.16}$$

If M_i is the model that corresponds to a diagnosis test δ_i, then we will motivated by the guidelines say that the model M_i is tested. Note that by definition of assump(M_i), it holds that

$$\text{sys} \in \text{assump}(M_i) \rightarrow \mathbf{z} \in \mathcal{O}(M_i) \tag{2.17}$$

This means that if the guidelines (a) and (b) are followed, then it holds that

$$T_i(\mathbf{z}) \in \mathcal{R}_i \rightarrow \mathbf{z} \notin \mathcal{O}(M_i) \rightarrow \text{sys} \notin \text{assump}(M_i) \rightarrow \text{sys} \notin \Phi_i \tag{2.18}$$

That is, when the test quantity is within the rejection region, we can draw the conclusion that sys $\in \Phi_i$ can not hold. This expression is equivalent to the requirement (2.7) so the design goal has been achieved. Note that if (2.15) holds instead of only (2.14), a stronger conclusion can in general be drawn in (2.18). As said above, (2.15) is therefore a better choice than (2.14).

Given a model M_i, there exists many methods to compute a test quantity T_i and a rejection region $\mathcal{R}_i$ that satisfy (2.10). One alternative is to eliminate all unknown variables in M_i to derive equations containing only known variables. An equation $a(\mathbf{z}) = 0$ derived from M_i satisfies

$$\mathbf{z} \in \mathcal{O}(M_i) \rightarrow a(\mathbf{z}) = 0 \tag{2.19}$$

and is said to be a *consistency relation*, also called *analytical redundancy relation* or *parity relation*. The consistency relation $a(\mathbf{z}) = 0$ can then used in the construction of a test quantity. Some works on the consistency relation based approach are (Staroswiecki and Comtet-Varga, 2001; Frisk and Åslund, 2005; Yu and Shields, 1997; Basseville and Nikiforov, 1993). Other alternatives that may be used are to make an observer based design (Frank, 1994; Kinneart, 1999; Nikoukhah, 1998; Persis and Isidori, 2001; Massoumnia et al., 1989) or to make an identification based design (Isermann, 1993). Test quantity construction is not the subject of this thesis, but some design methods will be exemplified in examples.

Example 2.8

For the water-tank example consider the diagnosis test δ_1 constructed in Example 2.7. The test quantity in (2.10) was derived from the equations $M_1 = \{e_2, e_3, e_6, e_7\}$, i.e. M_1, T_1, and Φ_1 satisfy (2.16), i.e. guideline (b). Furthermore, it holds that assump$(M_1) = \Phi_1$, i.e. guideline (a) is also fulfilled. Note that with the same T_1, $\mathcal{R}_1$, and Φ_1, the model M_1 could also have been chosen as $M_1 = \{e_3, e_6, e_7\}$ to fulfill both guidelines. This is the minimal equation set from which T_1 can be derived. Later in this chapter, minimal sets from which test quantities can be derived will be of special interest.

2.3 Diagnosis System

A diagnosis system using the principle of consistency-based diagnosis takes the observations and tries to conclude which behavioral modes that are consistent with the observations. The output of the diagnosis system is therefore a set of system behavioral modes that the diagnosis system claims to be consistent with the observations. This set will be denoted by C and the behavioral modes in C are said to be *candidates*. Formally a diagnosis system based on the principle of structured hypothesis tests can be defined as follows:

Definition 2.3 (Diagnosis System, Δ). *A* ***diagnosis system*** *is a set Δ of diagnosis tests, i.e. $\Delta = \{\delta_1, \delta_2, \ldots\}$ of diagnosis tests together with the procedure to form the set of candidates as*

$$C(\mathbf{z}) = \bigcap_{H_i^0 \textit{ rejected}} \Phi_i^C \tag{2.20}$$

According to the definition, a diagnosis system Δ defines a function from the set of all possible observations $\mathbf{z}$ to $\mathcal{P}(\mathbf{B})$, i.e.

$$\Delta : \mathbb{Z} \to \mathcal{P}(\mathbf{B})$$

The *candidates* are ideally equal to the diagnoses. In practice, it is not efficient to compute the candidates in accordance with (2.20). An efficient way to compute the candidates can be found in (Nyberg, 2006).

2.3.1 Strategies for Designing Diagnosis Systems

To design a diagnosis system consists of finding the set of diagnosis tests to be included, and also for each diagnosis test, a test quantity $T_i(\mathbf{z})$, a rejection region $\mathcal{R}_i$, and a null hypothesis H_i^0. We will here study two different strategies for finding these items. The first starts from a given set of null hypotheses H_i^0, and the second from the diagnosis model $\mathbb{M}$ of the system to be diagnosed.

2.3.2 Starting From Null Hypotheses

One way of starting the design of a diagnosis system is simply to decide which null hypotheses to test, and then construct a suitable test quantity and rejection region for each hypothesis test. One straightforward strategy is for example to have one diagnosis test for each of the system behavioral-modes. This is especially attractive when only single faults are considered. For example, if the possible system behavioral-modes are **NF**, **F1**, **F2**, and **F3**, then the four null hypotheses become

$$\begin{aligned} H_1^0 : \text{sys} &\in \Phi_1 = \{\mathbf{NF}\} \\ H_2^0 : \text{sys} &\in \Phi_2 = \{\mathbf{F1}\} \\ H_3^0 : \text{sys} &\in \Phi_3 = \{\mathbf{F2}\} \\ H_4^0 : \text{sys} &\in \Phi_4 = \{\mathbf{F3}\} \end{aligned}$$

To fulfill (2.7), it is suggested to follow the guidelines (a) and (b) above. The guidelines will then tell us how to choose M_i, namely any set such that (2.14) is fulfilled. The test quantity T_i and the rejection region $\mathcal{R}_i$ should then be selected to fulfill (b).

Example 2.9
Consider again the water-tank example and assume that we want to design a diagnosis test for the null hypothesis H_2^0 : sys = **NF**. The requirement (2.14) is fulfilled for $M_2 = M_{\mathbf{NF}}$. By eliminating the unknown variables using the `diffalg` package in Maple, we obtain

$$\begin{aligned} y_w - y_q^2 &= 0 \\ 2\dot{y}_q y_q - y_q + u &= 0 \end{aligned}$$

The first consistency relation is the one used in test quantity T_1 in Example 2.7 and therefore we will now use the second consistency relation. In the second consistency relation, there is an differentiated signal $\dot{y}_q$ that either can be approximated or eliminated introducing dynamics (Frisk and Åslund, 2005). If the latter alternative is chosen a so called *residual* r can be defined as

$$\dot{r} + \beta r = \frac{d}{dt}(y_q^2) - y_q + u \tag{2.21}$$

where $\beta > 0$ to ensure stability. A residual is a signal should be zero in the fault-free case. The above expression can then be written in a state-space form

$$\begin{array}{rcl} \dot{x} &=& -\beta(x + y_q^2) - y_q + u \\ r &=& x + y_q^2 \end{array} \tag{2.22}$$

without the derivative $\dot{y}_q$ as input. For a time t, a test is defined by selecting $T_2(t) = r(t)$ where r is defined in (2.22) and $\mathcal{R}_2 = \{x | x \notin [-0.1\, 0.1]\}$. An unknown initial value of x in (2.22) leads in general to a non-zero r in the fault free case. However r converge, by construction, exponentially to zero if sys = **NF**. Hence, if the influence on $r(t)$ of an incorrect initial condition is less than 0.1 at time t, then $M_2, T_2(t), \mathcal{R}_2$ fulfill (2.16).

2.3.3 Starting From the Model

The idea of this strategy is to start out from the model relations and investigate which relations that can be grouped together to form models possible to test in diagnosis tests (Blanke et al., 2003), (Cassar and Staroswiecki, 1997), (Pulido and Alonso, 2002) (Travé-Massuyès et al., 2001). The null hypothesis H_i^0 : $sys \in \Phi_i$ will then be chosen as Φ_i = assump(M_i). In this way the relation (14) will of course be fulfilled. Then the selection of the test quantity T_i and the rejection region $\mathcal{R}_i$ should follow (b).

One requirement is that assump(M_i) $\neq \emptyset$. If this requirement would not be fulfilled, it would hold that $\Phi_i^C = \mathbf{B}$. This means that the result of rejecting a null hypothesis would be that we can draw the conclusion sys $\in \mathbf{B}$, i.e. the test can never provide any information.

Definition 2.4 (Feasible)**.** *A set M of equations is **feasible** if* assump(M) $\neq \emptyset$.

Another requirement on the subset M_i is that there must be some $\mathbf{z}$ such that the relations M_i cannot all be fulfilled and this motivates the following definition.

Table 2.3: Rejectable models contained in (2.4).

i	Model M_i	Assumption $\Phi_i = \text{assump}(M_i)$
1	$\{e_3, e_6, e_7\}$	$\phi(T = \text{NF} \wedge W = \text{NF} \wedge Q = \text{NF})$
2	$\{e_1, e_2, e_3, e_7\}$	$\phi(P = \text{NF} \wedge T = \text{NF} \wedge Q = \text{NF})$
3	$\{e_1, e_2, e_6, e_7\}$	$\phi(P = \text{NF} \wedge W = \text{NF} \wedge Q = \text{NF})$
4	$\{e_1, e_2, e_3, e_6\}$	$\phi(P = \text{NF} \wedge T = \text{NF} \wedge W = \text{NF})$
5	$\{e_4, e_5, e_6, e_7\}$	$\phi(T = \text{C} \wedge W = \text{NF} \wedge Q = \text{NF})$
6	$\{e_3, e_6, e_8, e_9\}$	$\phi(T = \text{NF} \wedge W = \text{NF} \wedge Q = \text{B})$
7	$\{e_4, e_5, e_6, e_8, e_9\}$	$\phi(T = \text{C} \wedge W = \text{NF} \wedge Q = \text{B})$
8	$\{e_1, e_2, e_6, e_8, e_9\}$	$\phi(P = \text{NF} \wedge W = \text{NF} \wedge Q = \text{B})$
9	$\{e_1, e_2, e_4, e_5, e_7\}$	$\phi(P = \text{NF} \wedge T = \text{C} \wedge Q = \text{NF})$
10	$\{e_1, e_2, e_3, e_8, e_9\}$	$\phi(P = \text{NF} \wedge T = \text{NF} \wedge Q = \text{B})$
11	$\{e_1, e_2, e_4, e_5, e_6\}$	$\phi(P = \text{NF} \wedge T = \text{C} \wedge W = \text{NF})$
12	$\{e_1, e_2, e_4, e_5, e_8, e_9\}$	$\phi(P = \text{NF} \wedge T = \text{C} \wedge Q = \text{B})$

Definition 2.5 (Rejectable)**.** *A set M of equations is* ***rejectable*** *if*

$$\exists \mathbf{z} \in \mathbb{Z} \forall \mathbf{x} \in \mathbb{X} : \neg M(\mathbf{x}, \mathbf{z}) \tag{2.23}$$

Note that a feasible model M is rejectable if and only if

$$(\bigcup_{b \in \mathbf{B}} \mathcal{O}(M_b)) \setminus \mathcal{O}(M) \neq \emptyset \tag{2.24}$$

This means that there are possible observations that makes the model M inconsistent. If the model M_i is not a rejectable model, the test quantity would always be zero, or close to zero, and the test would make no sense.

The next definition is related to the notion of a rejectable model.

Definition 2.6 (Analytical Redundancy)**.** *There exists* ***analytical redundancy*** *in a model M if it is rejectable.*

This definition is in accordance with the definition of analytical redundancy in (Isermann and Balle, 1997). The question that remains is how to find subsets M_i that are feasible and rejectable. Given some natural assumptions about the model, the problem of finding suitable subsets M_i can often be solved by only studying the structural properties of the model. This is not the topic of this chapter but will be discussed later in Chapter 8.

Example 2.10

Now consider the water-tank system and assume that the subsets M_i with their corresponding assumptions assump(M_i) shown in Table 2.3 have been found to be rejectable models by studying the structural properties of the model. Exactly how these sets are computed are described in Example 8.11. As said above, Φ_i is then chosen as $\Phi_i = \text{assump}(M_i)$. By eliminating unknown variables with for example the `diffalg`-package in Maple, one consistency relation can be derived from each model M_i. These consistency relations are shown in Table 2.4. With these consistency relations, test quantities $T_i(\mathbf{z})$ and rejection regions $\mathcal{R}_i$ can be

Table 2.4: Consistency relations for the rejectable models in Table 2.3.

	Consistency Relation		
1	y_w	$=$	y_q^2
2	$\dot{y}_q$	$=$	$(-y_q + u)/(2y_q)$
3	$\dot{y}_w$	$=$	$-y_q + u$
4	$\dot{y}_w^2$	$=$	$y_w + 2\dot{y}_w u - u^2$
5	$\dot{y}_w$	$=$	$2\dot{y}_q y_w / y_q$
6	$\dot{y}_w^2$	$=$	$4\dot{y}_q^2 y_w$
7	$\ddot{y}_w$	$=$	$\dot{y}_w(\dot{y}_w \dot{y}_q + 2\ddot{y}_q y_w)/(2\dot{y}_q y_w)$
8	$\ddot{y}_w$	$=$	$-\dot{y}_q + \dot{u}$
9	$\ddot{y}_q$	$=$	$\dot{y}_q(y_q \dot{u} - \dot{y}_q u)/(y_q(-y_q + u))$
10	$\ddot{y}_q$	$=$	$(-4\dot{y}_q^2 - 4\dot{y}_q^3 + 2\dot{u}\dot{y}_q - \dot{y}_q + \dot{u})/(2u)$
11	$\ddot{y}_w$	$=$	$(\dot{y}_w^2 - \dot{y}_w u + 2\dot{u} y_w)/(2y_w)$
12	$\dddot{y}_q^2$	$=$	$-(4\dot{y}_q^2\ddot{y}_q^3 + 3u\ddot{u}\ddot{y}_q^3 + 4\dot{y}_q^2\ddot{u}\ddot{y}_q^2 - 3u\ddot{y}_q^4 - 6\dot{u}\dot{y}_q\ddot{y}_q^3 + \dot{y}_q^2\ddot{u}^2\ddot{y}_q - 3\dot{u}\dot{y}_q\ddot{u}\ddot{y}_q^2 -$ $2\dot{u}\dot{y}_q^2\dddot{y}_q\ddot{y}_q + 4\dddot{y}_q u\dot{y}_q\ddot{y}_q^2 - u\ddot{u}\dddot{y}_q\dot{y}_q\ddot{y}_q - 3u\dddot{y}_q\dot{u}\ddot{y}_q^2 - \dot{u}\dot{y}_q^2\ddot{u}\dddot{y}_q +$ $3\dot{u}^2\dot{y}_q\dddot{y}_q\ddot{y}_q)/(\dot{y}_q u(-\dot{y}_q + \dot{u}))$

constructed to fulfill (b). The complexity of the consistency relation derived from the set M_{12} indicates that an observer based approach might be a better alternative for this model.

2.4 Connection to FDI Methods

FDI methods presented in the literature, have focused mostly on residual generation and how disturbances and faults are to be decoupled. To use residuals is the most common way to construct test quantities within the field of FDI. The reason to decouple disturbances is to avoid false alarms, and the reason to decouple faults is to obtain residuals that are sensitive to different subsets of faults, so that isolation can be performed. From a residual r_i, a test quantity can for example be formed as $T_i = |r_i|$ or

$$T_i = \sum_{t=t_0}^{t=t_0+N} r_i^2(t) \tag{2.25}$$

Consider a linear system, typically found in FDI literature:

$$\dot{x} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u_a + \begin{bmatrix} 1 \\ 1 \end{bmatrix} d + \begin{bmatrix} 2 \\ 1 \end{bmatrix} f_1 + \begin{bmatrix} 1 \\ 0 \end{bmatrix} f_2 \tag{2.26a}$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} d \tag{2.26b}$$

where x is the dynamic state, u_a the actuator signal, y the sensor signals, and d an unknown disturbance signal. The signals f_1 and f_2 are used to model two different faults of the system and are non-zero only if the corresponding fault is present. The system itself is considered to be one component with three possible behavior modes: **NF**, **F1**, and **F2**. As seen, no actuator or sensor faults have been considered. A model $\boldsymbol{M}$ for this system, rewritten using the modeling framework suggested here is shown in Table 2.5.

Table 2.5: The model (2.26) rewritten using the modeling framework suggested here.

Assumption	Equation	Expression
	e_1	$\dot{x}_1 = x_1 + x_2 + d + 2\,f_1 + f_2$
	e_2	$\dot{x}_2 = x_1 + 2\,u_a + d + f_1$
	e_3	$y_1 = x_1$
	e_4	$y_2 = x_2$
	e_5	$y_3 = x_1 + x_2 + d$
sys $\in \{\mathbf{NF}, \mathbf{F2}\}$	e_6	$f_1 = 0$
sys $\in \{\mathbf{NF}, \mathbf{F1}\}$	e_7	$f_2 = 0$

The goal now is to find some residual for the system (2.26). In all residuals, the unknown disturbance d must be decoupled. To facilitate isolation, the goal is also to decouple different faults in different residuals. By linear-algebra manipulations of the system (2.26) (e.g. see (Nyberg and Frisk, 2006)), a number of residual generators can be found (here in the form of so called parity relations), for example:

$$\begin{aligned} r_1 &= -\dot{y}_1 + y_3 \\ r_2 &= 4u_a + \dot{y}_1 - 2y_2 - 2\dot{y}_2 + y_3 \\ r_3 &= 2u_a - y_2 - \dot{y}_2 + y_3 \end{aligned}$$

By carefully studying the formula of each residual, it can be realized that the sensitivity to the faults is according to the second column of Table 2.6.

A "0" means that when the behavioral mode of the column is present the residual of that row will be zero. An "X" means that the residual will be zero or non-zero. That is, in residual r_2, the fault signal f_1 has been decoupled, and in r_3, f_2 has been decoupled.

To see the relationship with the framework presented here, we have to investigate exactly which equations that have been used to form each residual. It turns out that to form residual r_1, i.e. to derive the equation $-\dot{y}_1 + y_3 = 0$, from the equations in the model $\boldsymbol{M}$, exactly the equations e_1, e_3, e_5, e_6, and e_7 have to be used. The equations M_i used to derive r_1, r_2, and r_3 can be seen in

Table 2.6: A decision structure.

	NF	**F1**	**F2**	M_i	assump(M_i)
r_1	0	X	X	$\{e_1, e_3, e_5, e_6, e_7\}$	$\{\mathbf{NF}\}$
r_2	0	0	X	$\{e_1, e_2, e_3, e_4, e_5, e_7\}$	$\{\mathbf{NF}, \mathbf{F1}\}$
r_3	0	X	0	$\{e_2, e_4, e_5, e_6\}$	$\{\mathbf{NF}, \mathbf{F2}\}$

the third column of the table. The assumptions for each equation set M_i, i.e. assump(M_i), can be seen in the fourth column of the table.

In conclusion, the FDI methods for residual generation, which can be based on e.g. parity relations or observers, can be utilized in the framework presented here. By keeping track of exactly which set M_i of equations that are used in the construction of each residual, the expression assump(M_i) can be obtained. This is then the only thing that is needed to facilitate fault isolation in the way proposed in this thesis.

2.5 Completeness and Soundness Properties of Diagnosis Systems

Ideally the output of the diagnosis system should specify exactly which the diagnoses are, i.e. the set of candidates $C(\mathbf{z})$ produced by the diagnosis system according to (2.20), should be equal to the set of diagnoses $\mathcal{D}(\mathbf{z})$ given any observation $\mathbf{z}$. However depending on how the diagnosis system is constructed, it is not sure that this exact relationship between the candidates and the diagnoses holds.

Example 2.11

Consider a diagnosis model $\mathbb{M}$ with the system behavioral modes $\mathbf{B} = \{\mathbf{NF}, \mathbf{F1}, \mathbf{F2}\}$ and with the model equations shown in Table 2.7 where y_i and u are known signals, x is an unknown signal, and f_i are unknown fault signals. By following the decoupling strategy presented in (Gertler, 1991) and (Mattone and Luca, 2006) in the construction of a diagnosis system,we would do the following diagonal design:

	NF	**F1**	**F2**	assump(M_i)
T_1	0	X	0	$\{\mathbf{NF}, \mathbf{F2}\}$
T_2	0	0	X	$\{\mathbf{NF}, \mathbf{F1}\}$

If p denotes the differential operator then the corresponding test quantities are

$$T_i = G_i(p)(py_i - u) = G_i(p)pf_i \tag{2.27}$$

where $G_i(p)$ for $i = \{1, 2\}$ are some filters chosen such that the transfer functions from y_i and u to T_i for $i = \{1, 2\}$ are proper and stable. Assume that both the rejection regions are $\mathcal{R}_1 = \mathcal{R}_2 = \mathbb{R} \setminus \{0\}$.

Assume that there is a fault **F1** with $f_1 = \sin(t)$ and consider the observation $u \equiv 0$, $y_1 = \sin(t)$, and $y_2 \equiv 0$. By substitution of the observed values for the known variables in $\{e_1, e_2, e_3\}$, we get that $f_1 = \sin(t)$ and $f_2 \equiv 0$. The equation e_4 cannot be true and this implies that the set of diagnosis is $\mathcal{D} = \{\mathbf{F1}\}$. If the initial conditions are assumed to be known, the test quantities become $T_1(t) = -G_1(p)\sin(t) \in \mathcal{R}_1$ and $T_2 \equiv 0 \notin \mathcal{R}$ according to (2.27). The candidate set is then $C = \{\mathbf{F1}\}$ according to (2.20). Hence for this observation the set of diagnoses is equal to the set of candidates, i.e. $\mathcal{D} = C$.

Now, assume instead that there is a fault **F1** with $f_1 \equiv 1$ and that the observations are $u \equiv 0$, $y_1 \equiv 1$, and $y_2 \equiv 0$. The set of diagnosis is again $\mathcal{D} = \{\mathbf{F1}\}$. However, the set of candidates are not the same. By substitution of the observed values for the known variables in (2.27), we get that $T_1 = T_2 \equiv 0$.

This implies that the candidates are $C = \{\mathbf{NF}, \mathbf{F1}, \mathbf{F2}\} \supset \mathcal{D}$. Hence for this observation, the set of diagnoses is a subset of the set of candidates. This means that the diagnosis system with the two test quantities in (2.27) does not use all available information in the diagnosis model in Table 2.7.

Table 2.7: The model (2.26) rewritten using the modeling framework suggested here.

Assumption	Equation	Expression
	e_1	$\dot{x} = u$
	e_2	$y_1 = x + f_1$
	e_3	$y_2 = x + f_2$
$\text{sys} \in \{\mathbf{NF}, \mathbf{F2}\}$	e_4	$f_1 = 0$
$\text{sys} \in \{\mathbf{NF}, \mathbf{F1}\}$	e_5	$f_2 = 0$

Next two definition concerning the relation between the diagnosis and the candidates are presented.

Definition 2.7 (Complete). *Given a diagnosis model* $\mathbb{M}$*, a diagnosis system* Δ *is* ***complete*** *with respect to* $\mathbb{M}$ *if*

$$\mathcal{D}(\mathbf{z}) \subseteq C(\mathbf{z}) \tag{2.28}$$

for all $\mathbf{z} \in \mathbb{Z}$*.*

Definition 2.8 (Sound). *Given a diagnosis model* $\mathbb{M}$*, a diagnosis system* Δ *is* ***sound*** *with respect to* $\mathbb{M}$ *if*

$$C(\mathbf{z}) \subseteq \mathcal{D}(\mathbf{z}) \tag{2.29}$$

for all $\mathbf{z} \in \mathbb{Z}$*.*

The candidates are the diagnoses if and only if the diagnosis system is sound and complete according to these definitions. The diagnosis system in Example 2.11 is not sound, because it was shown that there exists an observation that leads to $\mathcal{D} \subset C$.

2.5.1 Designing a Complete Diagnosis System

By computing the set of candidates as in (2.20) and following the two guidelines for designing tests, i.e. (2.15) and (2.16), the diagnosis system becomes complete as the next theorem shows.

Theorem 2.1. *Given a diagnosis model* $\mathbb{M}$*, a diagnosis system* Δ *is complete with respect to* $\mathbb{M}$ *if each test* $\delta_i \in \Delta$ *fulfills (2.15) and (2.16).*

Proof. By using (2.5), the equivalent expression

$$b \notin C(\mathbf{z}) \rightarrow \mathbf{z} \notin \mathcal{O}(M_b) \tag{2.30}$$

to (2.28) can be derived. From $b \notin C(\mathbf{z})$ and (2.20), it follows that there exists an $\delta_i \in \Delta$ such that $b \in \Phi_i$ and H_i^0 is rejected, i.e. $T_i(\mathbf{z}) \in \mathcal{R}$. This and (2.16) imply that

$$\mathbf{z} \notin \mathcal{O}(M_i) \tag{2.31}$$

From $b \in \Phi_i$ and (2.15), it follows that $b \in \text{assump}(M_i)$. From $b \in \text{assump}(M_i)$, (2.13), and (2.4) it follows that $M_i \subseteq M_b$. This and (2.31), imply that $\mathbf{z} \notin \mathcal{O}(M_b)$ which according to (2.30) completes the proof. □

This theorem tells that the completeness is easily obtained by following the two guidelines for designing tests. For Example 2.11 the two tests fulfill (2.15) and (2.16), i.e. the diagnosis system defined in the example is complete according to Theorem 2.1.

2.5.2 Designing a Sound Diagnosis System

A diagnosis system is sound if and only if for each behavioral mode $b \in \mathbf{B}$ and for each observation

$$\mathbf{z} \in \mathbb{Z} \setminus \mathcal{O}(M_b) \tag{2.32}$$

there exists a test δ_i such that the following expression holds

$$(T_i(\mathbf{z}) \in \mathcal{R}_i) \wedge (b \in \Phi_i) \tag{2.33}$$

This can normally not be obtained for noisy systems and a discussion about this is given in Section 2.7. However to start with, we will assume the ideal case with no noise. In Section 2.3.1 two strategies to design diagnosis systems were studied, one when starting from null hypotheses and one starting from the model. By using the first strategy there is only one possible choice that generally guarantees that a sound diagnosis system can be constructed and this choice is to have one test for each behavioral mode. Of course, there can be other sets of null hypotheses leading to a sound diagnosis system but this is dependent on the particular diagnosis model that is used. Therefore, we will study the second strategy, i.e. to start with the model to design a sound diagnosis system.

By using the strategy presented in Section 2.3.3, i.e. starting from a set models $\omega = \{M_1, M_2, \ldots\}$, it is natural to wonder if it is possible to construct a sound diagnosis system testing the consistency of the models in ω. This is answered in the next theorem.

Theorem 2.2. *Let $\mathbb{M}$ be a diagnosis model and $\omega = \{M_i\}$ a set of models. There exist tests δ_i testing the models $\omega = \{M_i\}$ and designed according to the guidelines (a) and (b) such that $\Delta = \{\delta_i\}$ is a sound diagnosis system with respect to a diagnosis model $\mathbb{M}$ if and only if*

$$\mathcal{O}(M_b) = \bigcap_{M_i \in \omega : M_i \subseteq M_b} \mathcal{O}(M_i) \tag{2.34}$$

for all $b \in \mathbf{B}$.

Before we prove the theorem, note that a diagnosis system with tests designed according to the guidelines (a) and (b) is also complete according to Theorem 2.1. Hence a diagnosis system designed according to the requirements given in Theorem 2.2 is a sound and complete diagnosis system with respect to $\mathbb{M}$.

Proof. First, we show existence of a sound diagnosis system under condition (2.34). Let the tests $\delta_i \in \Delta$ be ideal tests in the sense that $T_i(\mathbf{z}) \in \mathcal{R}_i^C$ if and only if $\mathbf{z} \in \mathcal{O}(M_i)$, i.e. equivalence holds in (2.16) and Φ_i is chosen according to (2.15). These tests clearly fulfill the guidelines (a) and (b). According to the construction of the tests Δ, the null hypothesis H_i^0 for a test $\delta_i \in \Delta$ and the observation sets $\mathcal{O}(M_i)$ are related as

$$H_i^0 \text{ rejected } \leftrightarrow \mathbf{z} \notin \mathcal{O}(M_i) \tag{2.35}$$

We show (2.29) by proving that for any $\mathbf{z} \in \mathcal{Z}$ and for any $b \notin \mathcal{D}(\mathbf{z})$ it follows that $b \notin \mathcal{C}(\mathbf{z})$. Let $\mathbf{z} \in \mathcal{Z}$ be an arbitrary observation and let the behavioral mode b be arbitrarily chosen such that $b \notin \mathcal{D}(\mathbf{z})$. From (2.5) it follows that

$$\mathbf{z} \notin \mathcal{O}(M_b) \tag{2.36}$$

Then, (2.34) implies that there exists a test $\delta_i \in \Delta$ such that

$$\mathbf{z} \notin \mathcal{O}(M_i) \tag{2.37}$$

and

$$M_i \subseteq M_b \tag{2.38}$$

The formulas (2.35) and (2.37) imply that H_i^0 is rejected. Furthermore, from (2.20) it follows that

$$\mathcal{C}(\mathbf{z}) \subseteq \Phi_i^C \tag{2.39}$$

The formulas (2.15) and (2.38) imply that $b \in \Phi_i$, which together with (2.39) implies that $b \notin \mathcal{C}(\mathbf{z})$, i.e. the diagnosis system is sound.

Now, we prove the opposite direction, i.e. if (2.34) does not hold for all behavioral modes, then no sound diagnosis system Δ exists.

Assume that (2.34) is not true for a behavioral mode $b \in \mathbf{B}$. The set $\mathcal{O}(M_b)$ cannot be a superset of the set defined in the right hand side of (2.34) and therefore the assumption implies that

$$\mathcal{O}(M_b) \subset \bigcap_{M_i \in \omega : M_i \subseteq M_b} \mathcal{O}(M_i) \tag{2.40}$$

Since (2.34) is not true for b it follows that there exists an observation $\mathbf{z}$ such that

$$\mathbf{z} \notin \mathcal{O}(M_b) \tag{2.41}$$

and

$$\mathbf{z} \in \mathcal{O}(M_i) \subseteq \mathcal{Z} \tag{2.42}$$

for all $M_i \in \omega$ such that $M_i \subseteq M_b$. The subset-relation in (2.42) follows from (2.3). The behavioral mode b is not a diagnosis according to (2.41), i.e. $b \notin \mathcal{D}(\mathbf{z})$. However, next we will prove that the behavioral mode $b \in \mathcal{C}(\mathbf{z})$, which implies that Δ is not sound. The behavioral mode b is a candidate if and only if H_i^0 is not rejected for all ϕ where $b \in \phi$ according to (2.20). From (2.15), it follows that $b \in \phi$ if and only if $M_i \subseteq M_b$. For any model $M_i \in \omega$ where $M_i \subseteq M_b$, (2.42) is true. This and (2.16) give that

$$T_i(\mathbf{z}) \notin \mathcal{R}_i \tag{2.43}$$

i.e. H_i^0 is not rejected. Since this is true for all tests δ_i where $b \in \phi$ it follows from (2.20) that $b \in \mathcal{C}(\mathbf{z})$ and the theorem follows. □

Condition (2.34) means that without making a diagnosis test based on M_b, tests for the models M_i and with equivalence in (2.16) can always be used to determine the consistency of M_b. This can also be formulated as that it is possible to determine the consistency of M_b by checking the consistency of the models M_i. In general there exist many solutions ω to (2.34), and we are particularly interested in sets with small cardinality, because the number of models in ω is equal to the number of tests in the diagnosis system based on ω. It is desirable to use few tests to reduce the online computational cost of running the diagnosis system.

Example 2.12
Continuation of Example 2.11. It has been concluded that the diagnosis system presented in Example 2.11 is not sound. However, if we add a test δ_3 such that $M_3 = \{e_2, e_3, e_4, e_5\}$, $\Phi_3 = \text{assump}(M_3) = \{\mathbf{NF}\}$, $T_3 = y_1 - y_2$, and $\mathcal{R}_3 = \mathbb{R} \setminus \{0\}$, then a sound and complete diagnosis system is obtained. The resulting decision structure is then:

	NF	**F1**	**F2**	assump(M_i)
T_1	0	X	0	$\{\mathbf{NF}, \mathbf{F2}\}$
T_2	0	0	X	$\{\mathbf{NF}, \mathbf{F1}\}$
T_3	0	X	X	$\{\mathbf{NF}\}$

By carefully studying the diagnosis model in Table 2.7, it follows from Theorem 2.2 that the minimum number of models that must be tested to obtain a sound diagnosis system is three. Hence the diagnosis system with T_1, T_2, and T_3 contains the least number of tests that any sound diagnosis system contains.

An upper bound for the number of models M_i needed to fulfill (2.34) for all $b \in \mathbf{B}$ is the number of system behavioral modes which corresponds to the straightforward solution of (2.34) to test all behavioral models, i.e. $\omega = \{M_b | b \in \mathbf{B}\}$. This corresponds to the strategy presented in Section 2.3.2. This strategy can handle models with a large number of faults when considering only single faults. However, when considering also multiple faults the number of behavioral modes grows exponentially with the number of components. Therefore this method runs into complexity problems when considering systems with a large number of components that can fail.

In the next section, it is shown that the number of tested models M_i need not be equal to the number of behavioral modes, because each model M_i can contribute to determining the consistency of a large number of behavioral models.

2.6 Testing Small Models

In this section an example illustrates how the number of tests can be decreased by testing models with few equations.

Consider the electrical circuit shown in Figure 2.3 consisting of a battery B, two resistors $R1$ and $R2$, one ideal voltage sensor $S1$, and two ideal current sensors $S2$ and $S3$. All six component have two behavioral modes, the no-fault mode NF and the unknown fault mode UF. The set consisting of the no-fault behavioral mode, all single faults, and all multiple faults is $\mathbf{B}$. The fault-free

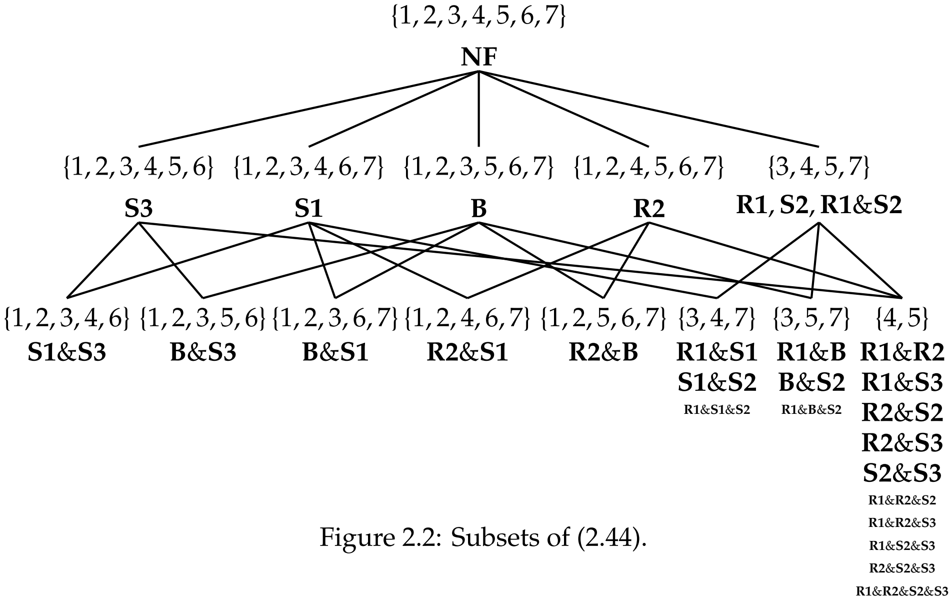

Figure 2.2: Subsets of (2.44).

behavior of the components are described by the model $\boldsymbol{M}$:

Assumption	Equation	Expression	
	(1)	$I - I_1 - I_2$	$= 0$
$R1 = \text{NF}$	(2)	$V - I_1 R_1$	$= 0$
$R2 = \text{NF}$	(3)	$V - I_2 R_2$	$= 0$
$B = \text{NF}$	(4)	$V - U$	$= 0$
$S1 = \text{NF}$	(5)	$V - y_V$	$= 0$
$S2 = \text{NF}$	(6)	$I - y_I$	$= 0$
$S3 = \text{NF}$	(7)	$I_2 - y_{I_2}$	$= 0$

(2.44)

where I, I_1, I_2 are currents; V the voltage across the battery; R_1 and R_2 are resistances which are assumed to be known constants; U is the expected voltage across the battery; and y_V, y_I, and y_{I_2} are measurements of V, I, and I_2 respectively. This means that

$$\begin{aligned} \mathbf{X} &= \{I, I_1, I_2, V\} \\ \mathbf{Z} &= \{U, y_V, y_{I_2}\} \end{aligned}$$

and the corresponding domains are $\mathbb{X} = \mathbb{R}^4$ and $\mathbb{Z} = \mathbb{R}^3$.

A straightforward way to fulfill (2.34) for all $b \in \mathbf{B}$ is as said before to test all behavioral models. For the electrical circuit, where all multiple faults are considered, there are $2^6 = 64$ behavioral modes. Next it will be discussed how to reduce the number of tests from the number of behavioral modes.

First, there are behavioral models that are not rejectable models. In the electrical circuit only 29 out of the 64 behavioral models are rejectable models. The 29 behavioral modes with rejectable behavioral models are those seen in Figure 2.2. This figure will below be explained more in detail.

There can be several rejectable behavioral models with equal observation sets, i.e. $\mathcal{O}(M_1) = \mathcal{O}(M_2)$ where M_1 and M_2 are two different behavioral models.

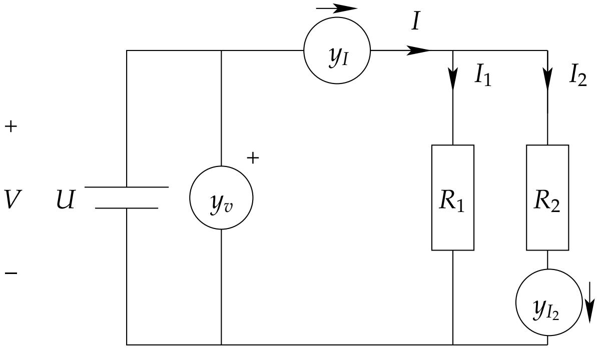

Figure 2.3: An electrical circuit

For the electrical circuit example the behavioral model $\{1, 4, 5, 6, 7\}$ of **R1&R2** and behavioral model $\{1, 2, 3, 4, 5\}$ of **S2&S3** have equal observation sets, i.e.

$$\mathcal{O}(\{1,4,5,6,7\}) = \mathcal{O}(\{1,2,3,4,5\}) = \{[U\ y_V\ y_I\ y_{I_2}] | U - y_V = 0\} \tag{2.45}$$

A minimal set of equations with the same observation set is $\{4, 5\}$ which is a subset of both the two behavioral models. It holds that

$$\mathcal{O}(\{1,4,5,6,7\}) = \mathcal{O}(\{1,2,3,4,5\}) = \mathcal{O}(\{4,5\})$$

Since the equation sets $\{4, 5\}$, $\{1, 4, 5, 6, 7\}$, and $\{1, 2, 3, 4, 5\}$ have equal observation sets, it is sufficient to check the consistency of for example only $\{4, 5\}$ to determine the consistency of both behavioral models. For each behavioral model in the example, it can be realized that there exists a unique minimal set with the same observation set. These equation sets and there corresponding behavioral modes are shown as a Hasse diagram in Figure 2.2 partial ordered by the subset relation. Instead of checking the consistency of all 29 rejectable behavioral models, it is sufficient to check the consistency of all the 14 models in the figure.

In the linear case it is also possible to determine the consistency of all models in the figure by checking the consistency of only the sets on the lowest levels. These 8 sets are the minimal sets that represents rejectable models. The constraint (2.34) for the behavioral modes on the lowest level imply that it is necessary to check all sets on the lowest level, except for $\{3, 4, 7\}$, $\{3, 5, 7\}$, and $\{4, 5\}$ which can be replaced by $\{1, 3, 4, 7\}$, $\{1, 3, 5, 7\}$, and $\{1, 4, 5\}$ correspondingly. Hence the minimum number of models that must be checked to obtain a sound and complete diagnosis system is 8. Hence this example shows that by testing small models the number of tests can be decreased.

2.7 Systems with Noise

The relation (2.7) can sometimes not hold strictly when the diagnosis test is used together with a noisy system. If noise is present, (2.7) can then be replaced by specifying the probability that (2.7) holds. In statistical hypothesis-testing theory, this requirement is usually written as

$$P(T_i(\mathbf{z}) \in \mathcal{R}_i | \text{sys} \in \Phi_i) \leq \alpha \tag{2.46}$$

That is, the probability of rejecting the null hypothesis $H_i^0 : \text{sys} \in \Phi_i$ given that $\text{sys} \in \Phi_i$ holds must be less or equal to a significance level α called the false alarm probability. The idea behind hypothesis testing is to have a false alarm probability that is very small, in fact so small that it is realistic to assume that the formula (2.7) holds.

The noise is the reason for why not equivalence is used in (2.7) and it can be realized as follows. Assume that a test δ_i fulfills (2.46), i.e. it is realistic to assume that the formula (2.7) holds. To get equivalence the following implication must hold

$$T_i \notin \mathcal{R} \rightarrow \text{sys} \in \Phi_i \tag{2.47}$$

i.e. in a statistical framework that

$$P(\text{sys} \in \Phi_i | T_i(\mathbf{z}) \notin \mathcal{R}_i) \approx 1 \tag{2.48}$$

However this is not likely to hold because $\mathcal{R}$ is chosen small to fulfill (2.46) and the probability in (2.48) increases with increasing $\mathcal{R}$. Hence in general it is only reasonable to assume that (2.46) can be fulfilled which motivates tests in the form (2.7).

In noisy (stochastic) systems, the model M is only approximate, or alternatively, is exact but includes stochastic terms. In this thesis only the first view is studied. A discussion of the second alternative is found in (Nyberg and Krysander, 2003).

The strategy to design a diagnosis test, that given an α fulfills (2.46), consists of two steps. The noise is neglected in the first step where a preliminary test is constructed by following the guidelines (a) and (b). In the second step the rejection region $\mathcal{R}'$ of the preliminary test is reduced such that (2.46) holds. Verification of (2.46) can be based on simulations of the test quantity $T_i(\mathbf{z})$ where the input is either obtained from Monte-Carlo simulations of the diagnosis model or directly obtained from measured data. The second step is also known as threshold selection.

Example 2.13

For the diagnosis model in Table 2.7, assume that the equation for sensor y_1 contains a noise term n representing white Gaussian distributed noise with zero mean and standard deviation σ according to

$$y_1 = x + f_1 + n \tag{2.49}$$

By neglecting the noise term, the test quantity $T_3 = y_1 - y_2$ has been derived in Example 2.12 from the model M_3. The test quantity is according to (2.49) equal to $T_3 = n$ when $f_1 = f_2 = 0$. A rejection region $\mathcal{R}_3 = \mathbb{R} \setminus [-J, J]$ described by a threshold J can now be selected such that the probabilistic correspondence (2.46) to (2.7) is fulfilled given a small false alarm probability α, i.e.

$$P(T_3 \in \mathcal{R}_i | \text{sys} \in \Phi_3) = P(|n| > J) \leq \alpha$$

The definition of soundness of a diagnosis system is mainly constructed for noise-free case. However if the noise distributions are bounded the definition of soundness can be useful also for models containing noise terms.

Example 2.14

Consider a diagnosis model with two behavioral modes **NF** and **UF** and a behavior of **NF** mode defined by the equation $y = n$ where y is a known variable and n is noise or uncertainty bounded to the interval $n \in [-1, 1]$. The diagnosis are

$$\mathcal{D}(y) = \begin{cases} \{\mathbf{NF}, \mathbf{UF}\} & \text{for } y \in [-1, 1] \\ \{\mathbf{UF}\} & \text{otherwise} \end{cases} \tag{2.50}$$

A sound and complete diagnosis system for this model is obviously defined by $\Delta = \{\delta_1\}$ where $T_1 = y$, $\Phi_1 = \{\mathbf{NF}\}$, and $\mathcal{R}_1 = \mathbb{R} \setminus [-1, 1]$.

If the support of the density function of the noise distribution is unbounded, we will take advantage of the two step approach for designing diagnosis systems. Since the noise is neglected in the first construction step, it is possible to require that the preliminary diagnosis system consisting of the preliminary tests should be sound. This requirement is relevant for the resulting diagnosis system for the noisy system and this can be motivated as follows.

In the limit when the noise level tends to zero, the preliminary rejection regions need not be reduced and soundness property of the preliminary diagnosis system is carried over to the resulting diagnosis system. Moreover, if the diagnosis system is not even sound in the ideal case, i.e. in the no-noise case, we can suspect that the diagnosis system in a noisy situation would be even worse, in the sense of (2.29).

In conclusion, noise is not explicitly included in the framework used here, nevertheless this section shows that the framework and the results presented here and in later chapters are also applicable to noisy systems. In particular completeness for diagnosis system designed for noisy systems can be obtained. Furthermore in the limit when the noise level tends to zero, soundness can be achieved and this is an upper bound of the diagnosis performance in terms of (2.29).

2.8 Conclusions

A new framework for model-based diagnosis has been presented. The isolation mechanism follows ideas from AI, namely to include in the model, how the validity of model equations depend on the presence of faults in each component. Isolation is then performed by propagating this information through the diagnosis system.

In contrast to AI, the diagnosis tests are computed off-line as in FDI. It has been shown in Section 2.4 how standard FDI methods, such as residuals based on parity relations or observers, can be used within the framework. In that case, the powerful isolation mechanism can be fully utilized.

Since the diagnosis tests used are really standard hypothesis tests from statistical hypothesis testing theory, it is possible to treat noise in a sound way. That is, even in a noisy system, faults can be correctly isolated.

In summary, the framework presented can efficiently handle: fault models, several different fault types (e.g. parameter- and additive faults), more than two behavioral modes per component, general differential-algebraic models, noise, uncertainties, decoupling of disturbances, static and dynamic systems, and isolation of multiple faults.

3

S D L
S S

Following the strategy described in Section 2.3.3, the construction of a diagnosis system starts with finding a set $\omega = \{M_1, \ldots, M_n\}$ of rejectable models to test. If the diagnosis system should be sound, these models $M_i \in \omega$ must fulfill

$$\mathcal{O}(M_b) = \bigcap_{M_i \subseteq M_b} \mathcal{O}(M_i) \tag{3.1}$$

for all $b \in \mathbf{B}$ according to Theorem 2.2.

In this chapter, the model equations are assumed to be linear static equations. The theory that will be developed for linear static equations will be reused for the linear

dynamic case in Chapter 6 and for structural methods handling the non-linear dynamic case in Chapter 8. In general there exist many solutions ω to (3.1), and we are particularly interested in rejectable sets with small cardinality according to the discussion in Section 2.6. Rank-conditions to check if a set ω fulfills (3.1) are developed. By using these conditions it is shown that it is sufficient to check the consistency of all minimal sets $M_i \subseteq M_b$ that are rejectable models. In the linear case such sets will also be called *minimal overdetermined (MO) set* of equations.

It is shown that the number of MO sets is dependent on the degree of redundancy, i.e. the number of linearly independent consistency relations. For a fixed order of redundancy, the number of MO sets is shown to be at most polynomial in the number of equations. Furthermore, it is shown that the degree of redundancy is limited by the number of sensors. In many applications, sensors are expensive and thus the redundancy degree is low even if the models contains a large number of components.

The main problem to solve in this chapter is how to find a set ω of models to test such that (3.1) is fulfilled for all behavioral modes $b \in \mathbf{B}$. This problem can be divided into a number of sub-problems, one for each behavioral mode in $\mathbf{B}$. Sections 3.1-3.7 only consider the sub-problem of finding an ω given a behavioral model M_b. Having solved these sub-problems, the solutions can be combined to solve the full problem for all behavioral modes in **B** and this is done in Section 3.8 and 3.9.

In Sections 3.1 we will study definitions presented in Chapter 2 such as, analytical redundancy, observation set, and rejectable models in the linear static case. The condition (3.1) is then analyzed in Sections 3.2 and 3.3. Section 3.4 states some important results, i.e. two rank conditions that can be used to determine if a given ω fulfills (3.1), and that it is sufficient to test all minimal sets of equations that are rejectable models, i.e. all MO sets. It is in general not necessary to test all MO sets and Section 3.5 gives some results of how to select a subset of all MO sets that fulfills (3.1). Section 3.6 extends the model-class considered to general linear static equations. In Section 3.7 the dependencies between the number of sensors, the redundancy, and the number of MO sets are studied.

Section 3.8 extends the result that it is sufficient to include all MO sets in ω to fulfill (3.1) for one behavioral mode to the case when considering all behavioral modes $b \in \mathbf{B}$. As in the case when considering only one behavioral mode, it is not necessary in general to include all MO sets in ω. In Section 3.9 algorithms are given that finds the minimum cardinality sets of ω containing MO sets that satisfy (3.1) for all behavioral modes $b \in \mathbf{B}$. The size of a minimum cardinality solution ω is equal to the minimum number of test that must be used to construct a sound diagnosis system. Several examples illustrate how a minimal number of MO sets, i.e. a minimal number of tests in the resulting diagnosis system, is computed. Finally fault sensitivity of residual generators based on MO models is discussed in Section 3.10 before the conclusions are drawn.

3.1 Linear Static Models

Consider a linear static model M_b for a specific behavioral mode:

$$H_b x + L_b z = 0 \tag{3.2}$$

where H_b and L_b are constant matrices, x is a vector of unknowns and z is a vector of known variables.

Example 3.1

Throughout this chapter we will use the electrical circuit example presented in Section 2.6 to illustrate concepts and theoretical results. Two behavioral modes **NF** and **R1&R2** and there corresponding behavioral models will be studied. By using the model (2.44), the behavioral model $M_{\mathbf{NF}}$ can be written in the matrix form (3.2) as

$$\overbrace{\begin{bmatrix} 0 & 1 & -1 & -1 \\ 1 & 0 & -R_1 & 0 \\ 1 & 0 & 0 & -R_2 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}^{H} \begin{bmatrix} V \\ I \\ I_1 \\ I_2 \end{bmatrix} + \overbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}}^{L} \begin{bmatrix} U \\ y_V \\ y_I \\ y_{I_2} \end{bmatrix} = 0 \tag{3.3}$$

The behavioral model for behavioral mode **R1&R2** is equal to the set $M_{\mathbf{R1\&R2}} = \{1, 4, 5, 6, 7\}$ of equations in (2.44).

To write for example the equations in $M_{\mathbf{R1\&R2}}$ in the form (3.2) by using the matrices H and L in (3.3), some matrix notation is needed. For a matrix A, an ordered *row index set* R and an ordered *column index set* C are defined such that $A = (A_{ij} | i \in R, j \in C)$, where A_{ij} is the (i, j)-entry of A. For $I \subseteq R$ and $J \subseteq C$, $A[I, J] = (A_{ij} | i \in I, j \in J)$ denotes the sub-matrix of A with row index set I and column index set J. Shorthand notations for the matrices $A[I, C]$ and $A[R, J]$ will be $A[I]$ and $A[:, J]$ respectively. Now, the set $M_{\mathbf{R1\&R2}}$ of equations can be written in the form (3.2) as

$$H[M_{\mathbf{R1\&R2}}]\boldsymbol{x} + L[M_{\mathbf{R1\&R2}}]\boldsymbol{z} = 0 \tag{3.4}$$

where the matrices H and L are defined in (3.3).

We will find sets ω of models M_i such that (3.1) is fulfilled for a behavioral mode b. That is the consistency of the models in ω will determine the consistency of the behavioral models M_b. In (3.1) observation sets are used and in the next section we will discuss consistency and observation sets in the linear static case.

3.1.1 Observation Sets

For linear static models an observation $\boldsymbol{z}$ is assumed to be a snap-shot of the vector $\boldsymbol{z}$, i.e. a value of the vector $\boldsymbol{z} = z_0 \in \mathbb{R}^{n_z}$ where n_z is the dimension of $\boldsymbol{z}$. Let n_x be the dimension of $\boldsymbol{x}$. A linear model

$$H\boldsymbol{x} + L\boldsymbol{z} = 0 \tag{3.5}$$

consisting of the equations M is *consistent* with an observation $\boldsymbol{z} = z_0$, if

$$\exists \boldsymbol{x} \in \mathbb{R}^{n_x}; H\boldsymbol{x} + Lz_0 = 0 \tag{3.6}$$

is true. The *observation set* $\mathcal{O}(M)$ for the equations M is then formally defined as

$$\mathcal{O}(M) = \{\boldsymbol{z} \in \mathbb{R}^{n_z} | \exists \boldsymbol{x} \in \mathbb{R}^{n_x}; H\boldsymbol{x} + L\boldsymbol{z} = 0\} \tag{3.7}$$

The observation set $\mathcal{O}(M)$ can in the linear case be expressed without $\boldsymbol{x}$ as follows. Let N_H be any matrix such that the rows of N_H is a basis for the left null-space of the matrix H. This means that N_H has the maximum independent rows which solves

$$N_H H = 0 \tag{3.8}$$

By multiplying (3.5) from left with N_H, we get

$$N_H L\boldsymbol{z} = 0 \tag{3.9}$$

The expression (3.6) is equivalent to (3.9), i.e.

$$\mathcal{O}(M) = \{\boldsymbol{z} \in \mathbb{R}^{n_z} | N_H L\boldsymbol{z} = 0\} \tag{3.10}$$

This result will be shown analogously for linear differential equations in Theorem 6.2. Each row of $N_H L$ defines a consistency relation, i.e. an equation containing only known variables. We will say that consistency relations are linearly independent if their corresponding rows in $N_H L$ are linearly independent.

3.1.2 Redundancy and Rejectable Models

Existence of redundancy was defined in the previous chapter, and for linear systems, it is also possible to quantify redundancy as follows.

Definition 3.1 (Redundancy of a Model)**.** *Let $\varphi : 2^{\mathbf{M}} \to \mathbb{N}$ be a function, from the family of subsets M in a set $\mathbf{M}$ of linear equations (3.5) to the natural numbers, defined by*

$$\varphi\,(M) = \text{rank}\,([H[M]\,L[M]]) - \text{rank}\,(H[M]) \tag{3.11}$$

*This number $\varphi\,(M)$ is the **redundancy of** M.*

Before redundancy is further discussed, a convenient matrix notation will be defined . Let A be a matrix with a row index set R and let B be a matrix with a row index set $I \subseteq R$ such that $B = A[I]$. Assume that the matrix $N = N_B$ has a row index set R' and column index set I. Then, let $N_{A[I]}$ be a matrix with the row index set R' and the column index set R such that the entries are defined by

$$(N_{A[I]})_{ij} := \begin{cases} N_{ij} & \text{if } i \in R', j \in I \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

where N_{ij} is the (i, j)-entry of N. From the definition $N_{A[I]}$, it follows that $N_{A[I]}[:, I] = N$, $N_{A[I]}[:, R \setminus I] = 0$, and

$$N_{A[I]}A = 0 \tag{3.13}$$

That is, the matrix $N_{A[I]}$ is a basis for the left null-space of the matrix $A[I]$ and zero-padded such that multiplication of the matrix A according to (3.13) is well defined.

Now, we will discuss Definition 3.1. Note that $\varphi\,(M) \geq 0$ for any set M and $\varphi\,(\emptyset) = 0$. The redundancy of a set M is equal to the maximum number of linearly independent consistency relations that can be derived from M, i.e.

$$\varphi\,(M) = \text{rank}\,(N_{H[M]}L) \tag{3.14}$$

where the number of columns in $N_{H[M]}$, according to the introduced notation, is equal to the number of rows in L. For a linear static model M, existence of redundancy according to Definition 2.6 is equivalent to $\varphi\,(M) > 0$. A linear static model M is a rejectable model if and only if M has redundancy, i.e. $\varphi\,(M) > 0$. Linear static models with redundancy are said to be *overdetermined* according to the next definition.

Definition 3.2 (Overdetermined Set)**.** *A set M of equations (3.5) is an **overdetermined set** if its redundancy is positive, i.e $\varphi\,(M) > 0$.*

A linear set of equations is a rejectable model if and only if it is an overdetermined set. The overdetermined models $M \subseteq M_{\mathbf{NF}}$ for the electrical circuit example are all the supersets to any of the models on the lowest level in Figure 2.2.

To conclude this section, behavioral modes $b \in \mathbf{B}$ with not overdetermined behavioral models fulfill (3.1) trivially. Next we will discuss which subsets ω of models that fulfill (3.1) for overdetermined behavioral models.

3.2 A Proper Overdetermined Set is Sufficient for Soundness

There can be equations in an overdetermined model that does not contribute to define the observation set of the model as the next example shows.

Example 3.2
Consider for the electrical circuit example the behavioral model $\{1, 4, 5, 6, 7\}$ of **R1&R2** and its observation set

$$\mathcal{O}(\{1, 4, 5, 6, 7\}) = \{[U\, y_V\, y_I\, y_{I_2}] | U - y_V = 0\} \tag{3.15}$$

defined by the consistency relation $U - y_V = 0$. A minimal set of equations that can be used to derived this consistency relation is $\{4, 5\}$ which is a subset of the behavioral model. It holds that

$$\mathcal{O}(\{1, 4, 5, 6, 7\}) = \mathcal{O}(\{4, 5\})$$

Since the equation sets $\{4, 5\}$ and $\{1, 4, 5, 6, 7\}$ have equal observation sets, it is sufficient to check the consistency of for example only $\{4, 5\}$ to determine the consistency of the behavioral model.

A set of equations, like the set $\{4, 5\}$, that has an observation set that no proper subsets have, will be called a *proper overdetermined set* and is defined by using the redundancy function as follows.

Definition 3.3 (Proper Overdetermined Set). *An overdetermined set M of equations (3.5) is a **proper overdetermined (PO) set** if*

$$\varphi(E) < \varphi(M) \tag{3.16}$$

for all $E \subset M$.

Note that a PO set M is a minimal set with redundancy $\varphi(M)$. For the example, the set $\{4, 5\}$ is, as said before, a PO set because

$$\varphi(\{4, 5\}) = 1 > 0 = \varphi(\{4\}) = \varphi(\{5\})$$

The difference between PO models and all other overdetermined models is that the removal of any equation in a proper overdetermined model will decrease the redundancy and therefore increase the dimension of the set of consistent observations defined by the remaining equations. The next theorem shows the relation between observation sets and PO sets.

Theorem 3.1. *An overdetermined set M of equations (3.5) is a PO set if and only if*

$$\mathcal{O}(M) \subset \mathcal{O}(E) \tag{3.17}$$

for all $E \subset M$.

Proof. Let E' be any set such that $E' \subset M$. It holds that

$$\mathcal{O}(M) \subseteq \mathcal{O}(E') \tag{3.18}$$

Proper inclusion in (3.18) as in (3.17) is obtained if and only if

$$\dim\mathcal{O}(M) < \dim\mathcal{O}(E') \tag{3.19}$$

Let the number of known variables in M be denoted n. Then it holds that

$$\dim\mathcal{O}(E) + \varphi(E) = n$$

for any set $E \subseteq M$. This implies that (3.19) holds if and only if

$$\varphi(M) > \varphi(E') \tag{3.20}$$

From the equivalence between proper inclusion in (3.18) and (3.20) and since $E' \subset M$ was arbitrarily chosen the theorem follows. □

For the behavioral model $\{1,4,5,6,7\}$ we have shown that there exists a PO set $\{4,5\}$ with exactly the same observation set and the next theorem states that for any model, and therefore also for any behavioral model, there exists a PO set with the same observation set.

Theorem 3.2. *If M is an overdetermined set of equations (3.5), then there exists at least one PO set $E \subseteq M$ such that*

$$\mathcal{O}(E) = \mathcal{O}(M) \tag{3.21}$$

Proof. If M is not minimal with $\mathcal{O}(M)$, then there exists at least one subset E that is minimal and fulfills (3.21). From Theorem 3.1, it then follows that E is a PO set. □

A consequence of this theorem is that it is sufficient to test PO sets to determine the consistency of any behavioral model.

Example 3.3

The PO sets in (2.44) are the sets shown in Figure 2.2. Note that **R1&R2** is written below $\{4,5\}$ to denote that $\{4,5\}$ has the same observation set as the behavioral model for **R1&R2**. Considering another behavioral model, i.e. the no-fault behavioral model, it turns out that this model is a PO set.

3.3 Properties of PO sets

We have shown that there exists a PO set M in any behavioral model M_b with equal observation set as the behavioral model. In the next section we will show that it is sufficient to check the consistency of the PO sets $M_i \subseteq M$ with redundancy one to determine the consistency of the PO set M and therefore also to determine the consistency of the behavioral model M_b. However to show this, some preliminary results about PO sets, presented in this section, need to be developed. This section will be more technical than the other sections in this chapter. One possible option can be to omit this section for now and read parts of it when results in this section are referenced in the following sections.

The first two theorems show properties of the redundancy-function.

Theorem 3.3. *If M is a set of equations (3.5) and M' an arbitrary subset $M' \subseteq M$, then it follows that*

$$\varphi(M') \leq \varphi(M) \tag{3.22}$$

The next lemma will be used to prove this theorem. Let A be a matrix and let R be the row sets of its rows and C the row sets of its columns. Then the following functions are defined:

$$\begin{aligned} \rho(I) = &\quad \text{rank}\,(A[I]) \qquad I \subseteq R \qquad (3.23) \\ \lambda(I,J) = &\quad \text{rank}\,(A[I,J]) \quad I \subseteq R, J \subseteq C \qquad (3.24) \end{aligned}$$

The next lemma is given and proved in (Murota, 2000).

Lemma 3.1. *If A is a matrix and $I_1, I_2 \subseteq R$ and $J_1, J_2 \subseteq C$ then*

$$\begin{aligned} \rho(I_1) + \rho(I_2) &\geq \rho(I_1 \cap I_2) + \rho(I_1 \cup I_2) \qquad (3.25) \\ \lambda(I_1, J_1) + \lambda(I_2, J_2) &\geq \lambda(I_1 \cup I_2, J_1 \cap J_2) + \lambda(I_1 \cap I_2, J_1 \cup J_2) \qquad (3.26) \end{aligned}$$

Inequality (3.25) means that the rank-function is a *sub-modular function* (Murota, 2000) on the family of subsets of R. Inequality (3.26) will now be used to prove Theorem 3.3.

Proof. Let $\begin{bmatrix} H & L \end{bmatrix}$ be A in Lemma 3.1 and let $I_1 = M$, $I_2 = M'$, J_1 be equal to the column set of $[H\,L]$, and J_2 be equal to the column set of H in (3.26). Then we obtain

$$\text{rank}\,(\begin{bmatrix} H & L \end{bmatrix}) + \text{rank}\,(H[M']) \geq \text{rank}\,(H) + \text{rank}\,(\begin{bmatrix} H[M'] & L[M'] \end{bmatrix})$$

By reordering the terms and by using (3.11) the inequality (3.22) follows. □

The next theorem reveals a redundancy pattern for the PO sets in a graph of the type shown in Figure 2.2 and also particularly identifies those with redundancy one.

Theorem 3.4. *If M is a PO set of equations (3.5) and $e \in M$, then*

$$\varphi\,(M \setminus \{e\}) = \varphi\,(M) - 1 \qquad (3.27)$$

Before we prove the theorem, the result is exemplified. A consequence of the theorem is that the PO sets can be organized in levels corresponding to their redundancy.

Example 3.4

In Figure 2.2 the PO sets on the lowest level have redundancy one and PO on the next level have redundancy two and so on. The no-fault behavioral model $\{1, 2, 3, 4, 5, 6, 7\}$ has therefore redundancy three because it is on the third level.

Now, we prove Theorem 3.4.

Proof. From Definition 3.3, it follows that

$$\varphi\,(M \setminus \{e\}) < \varphi\,(M)$$

Hence it remains to prove that

$$\varphi\,(M) - 1 \leq \varphi\,(M \setminus \{e\})$$

The left-hand side is equal to

$$\varphi\,(M) - 1 = \text{rank}\,([H\,L]) - \text{rank}\,(H) - 1$$

which can be estimated by using that

$$\text{rank}\,([H\,L]) - 1 \leq \text{rank}\,([H[M \setminus \{e\}]\,L[M \setminus \{e\}]])$$

and that

$$-\text{rank}\,(H) \leq -\text{rank}\,(H[M \setminus \{e\}])$$

as

$$\varphi\,(M) - 1 \leq \text{rank}\,([H[M \setminus \{e\}]\,L[M \setminus \{e\}]]) - \text{rank}\,(H[M \setminus \{e\}]) = \varphi\,(M \setminus \{e\})$$

This completes the proof. □

The following theorem is used to characterize PO sets.

Theorem 3.5. *A set M of equations (3.5) is a PO set if and only if the matrix $[H\,L]$ has full row-rank and*

$$\text{rank}\,(H[M \setminus \{e\}]) = \text{rank}\,(H) \tag{3.28}$$

for all $e \in M$.

Proof. Let M be a PO set and assume that $[H\,L]$ has not full row-rank. Let $H_i = H[M \setminus \{i\}]$, $L_i = L[M \setminus \{i\}]$, and $M_i = M \setminus \{i\}$ for any $i \in M$. Then there exists an $i \in M$ such that

$$\text{rank}\,([H\,L]) = \text{rank}\,([H_i\,L_i])$$

This and that $\text{rank}\,(H) \geq \text{rank}\,(H_i)$ imply that

$$\varphi\,(M_i) = \text{rank}\,([H_i\,L_i]) - \text{rank}\,(H_i) \geq \text{rank}\,([H\,L]) - \text{rank}\,(H) = \varphi\,(M) \tag{3.29}$$

i.e. M is not a PO set. Hence the matrix $[H\,L]$ has full row-rank and

$$\text{rank}\,(H) = \text{rank}\,([H\,L]) - \varphi\,(M) = |M| - \varphi\,(M) \tag{3.30}$$

Since M is a PO set, it holds according to Theorem 3.4 that

$$\varphi\,(M) - 1 \geq \varphi\,(M_i)$$

This implies that the right-hand side of (3.30) can be limited from above as

$$(|M| - 1) - (\varphi\,(M) - 1) \leq \text{rank}\,([H_i\,L_i]) - \varphi\,(M_i) = \text{rank}\,(H_i) \tag{3.31}$$

and (3.28) follows.

The converse direction is shown by using that $[H\,L]$ has full row-rank and (3.28) as

$$\varphi\,(M_i) = \text{rank}\,([H_i\,L_i]) - \text{rank}\,(H_i) < \text{rank}\,([H\,L]) - \text{rank}\,(H) = \varphi\,(M) \tag{3.32}$$

This and Theorem 3.3 imply that $\varphi\,(E) < \varphi\,(M)$ for each $E \subset M$, i.e. M is a PO set and this completes the proof. □

The condition (3.28) means that any row $e \in M$ can be written as a linear combination of the other rows in H. Let M be a PO set of equations defined by the matrices H and L. Then $[H\,L]$ has full row-rank and the redundancy of any subset $M' \subseteq M$ can therefore be simplified as

$$\varphi\,(M') = \text{rank}\,([[H[M']\,L[M']]]) - \text{rank}\,(H[M']) = |M'| - \text{rank}\,(H[M']) \tag{3.33}$$

Note that the simplified redundancy expression does not depend on the matrix L.

An important consequence of this theorem is that if M is a PO set then the non-zero columns in any basis $N_{H[M]}$ are equal to M, and this will be shown in the next example and corollary.

Example 3.5
Consider for the electrical circuit example the PO set $M_1 = \{4, 5\}$ and the set $M_2 = \{1, 4, 5, 6, 7\}$ which is not a PO set. For both these sets a corresponding basis $N_{H[M_i]}$ is

$$\begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 0 & 0 \end{bmatrix} \tag{3.34}$$

The non-zero columns are in this case $\{4, 5\}$ and this is a subset of both M_1 and M_2 and this holds also generally. Note that for the PO set M_1, the non-zero columns are equal to M_1 but this is not the case for the set M_2.

The next corollary shows that the non-zero columns of any basis for the left null space of $H[M]$ are exactly the columns M if M is a PO set.

Corollary 3.1. *If M is a PO set of equations (3.5) and N_H is a basis for the left null space of H, then it follows that no column in N_H is zero.*

Proof. From Theorem 3.5, it follows that any row in H can be written as a linear combination of the other rows, i.e. there exists a row-vector γ such that

$$\gamma H = 0 \tag{3.35}$$

and

$$\gamma[\{i\}] = 1 \tag{3.36}$$

The equation (3.35) implies that γ belongs to the left null space of H, i.e. there exists a row-vector γ' such that

$$\gamma = \gamma' N_H$$

This and (3.36) imply that the i:th column of N_H cannot be zero. Since $i \in M$ was arbitrarily chosen, the corollary follows. □

To explain a related result of Theorem 3.5 and this corollary, consider a set of equations (3.5) where $[H\,L]$ has full row rank and let the non-zero columns in any basis N_H for the left null space of H be denoted by M'. Any row in $H[M']$ can be written according to N_H as a linear combination of the other rows in $H[M']$. Theorem 3.5 then implies that M' is a PO set. This result is formally proved in Theorem 4.2 and will in Chapter 4 be used to compute PO sets.

3.4 Minimal Overdetermined Sets are Sufficient for Soundness

In Section 3.2, we have shown that there exists a PO set M in M_b with equal observation set, i.e. it is sufficient to check the consistency of one of these PO sets in M to determine the consistency of M. In this section a rank conditions will be developed to test if a given set $\{M_i | M_i \subseteq M\}$ of PO models can be used to determine the consistency of M. In particular, it is shown that it is sufficient to check the consistency of the PO sets $M_i \subseteq M$ with redundancy one to determine the consistency of the PO set M and therefore also to determine the consistency of the behavioral model M_b.

3.4.1 Defining and characterizing MO sets

The next definition and lemma introduces and characterizes the PO sets with redundancy one.

Definition 3.4 (Minimal Overdetermined Set)**.** *An overdetermined set M of equations (3.5) is a* ***minimal overdetermined (MO) set*** *if no proper subset is an overdetermined set.*

The next lemma relates MO sets and PO sets.

Lemma 3.2. *A set M of equations (3.5) is an MO set if and only if M is a PO set with $\varphi(M) = 1$.*

Proof. Assume that M is an MO set, i.e. $\varphi(M) > 0$ and $\varphi(E) = 0$ for all $E \subset M$. This implies that M is a minimal set with redundancy $\varphi(M)$, i.e. M is a PO set. From Theorem 3.4, it follows that $\varphi(M) = \varphi(M \setminus \{e\}) + 1 = 1$ for any $e \in M$.

Assume that M is a PO set with $\varphi(M) = 1$. This means according to Definition 3.3 that $\varphi(E) = 0$ for all $E \subset M$, i.e. M is an MO set. □

Note also that a set is an MO set if and only if the set is minimal and a PO set. According to the characterization of MO sets described in Lemma 3.2 it follows that the PO sets on the lowest level in Figure 2.2 are the MO sets.

3.4.2 Motivating Example for Using MO Sets

Next an example will be used to show that the consistency of PO sets can be determined by checking the consistency of MO sets.

Example 3.6

Consider the PO set $\{3, 4, 5, 7\}$ with redundancy 2. The MO sets contained in $\{3, 4, 5, 7\}$ are the three subsets $\{3, 4, 7\}$, $\{3, 5, 7\}$, and $\{4, 5\}$. The matrices $N_{H[M]}L$ corresponding the PO sets are:

$$\begin{array}{lll}
\text{PO set } M & N_{H[M]}L & \text{redundancy } \varphi(M) \\
\hline
\{3,4,5,7\} & \begin{bmatrix} -1 & 0 & 0 & R_2 \\ 0 & -1 & 0 & R_2 \end{bmatrix} & 2 \\
\{3,4,7\} & \begin{bmatrix} -1 & 0 & 0 & R_2 \end{bmatrix} & 1 \\
\{3,5,7\} & \begin{bmatrix} 0 & -1 & 0 & R_2 \end{bmatrix} & 1 \\
\{4,5\} & \begin{bmatrix} 1 & -1 & 0 & 0 \end{bmatrix} & 1
\end{array} \tag{3.37}$$

First, note that all rows corresponding to the MO sets are pairwise linearly independent, i.e. pairwise their consistency relations are linearly independent. Second, all rows corresponding to all MO sets belong to the space spanned by the rows corresponding to the set $\{3, 4, 5, 7\}$ with 2 linearly independent consistency relations. These two facts imply that any two rows corresponding to MO sets span both rows corresponding to $\{3,4,5,7\}$. Hence by checking the consistency of any two MO models the consistency of $\{3,4,5,7\}$ can be determined.

The concept of linearly independent consistency relations is central in the discussion here. The redundancy of a set M was in Section 3.1.2 seen to be equal to the maximum number of linearly independent consistency relations that can be derived from M. From any pair of MO sets in (3.37), two linearly independent consistency relations can be obtained. This motivates that the notion of redundancy and the closely related concept of observation set is possible to generalize to a family of equation set and this will be done next.

3.4.3 Observation Set for a Family of Equation Sets

By introducing redundancy of family of equation sets, it will be shown later in Theorem 3.6 that sets ω that satisfy (3.1) can be characterized by the generalized redundancy. This result is approached in this section by analyzing the right-hand side of (3.1) which defines the observation set for a set of models.

Given a set of linear equations M, consider a family of subsets $\omega = \{M_1, M_2, \ldots, M_n\}$ where $M_i \subseteq M$ for all $M_i \in \omega$. The task of checking the consistency of the set of models $M_i \in \omega$ is to evaluate, given an observation z, if

$$\exists x_i H[M_i]x_i + L[M_i]z = 0 \tag{3.38}$$

for all $M_i \in \omega$. In terms of observation sets this could be expressed as to check if z belongs to the set

$$\bigcap_{M_i \in \omega} \mathcal{O}(M_i) = \bigcap_{M_i \in \omega} \{z \in \mathbb{R}^{n_z} | \exists x_i \in \mathbb{R}^{n_x}; H[M_i]x_i + L[M_i]z = 0\} \tag{3.39}$$

Note that checking the consistency of the models in ω is not equivalent with checking the consistency of $\cup_{M_i \in \omega} M_i$. This follows from the fact that the set $\mathcal{O}(\cup_{M_i \in \omega} M_i)$ is equal to the set of $z \in \mathbb{R}^{n_z}$ consistent with (3.38) when requiring in addition that $x_i = x_j$ for all pairs of $M_i, M_j \in \omega$. In general, it holds that

$$\mathcal{O}(\cup_{M_i \in \omega} M_i) \subseteq \cap_{M_i \in \omega} \mathcal{O}(M_i) \tag{3.40}$$

The set of n linear matrix equations in (3.38) can be written in the standard form (3.5) as

$$\overbrace{\begin{bmatrix} H[M_1] & 0 & \cdots & 0 \\ 0 & H[M_2] & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & H[M_n] \end{bmatrix}}^{H_e} \overbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}^{x_e} + \overbrace{\begin{bmatrix} L[M_1] \\ L[M_2] \\ \vdots \\ L[M_n] \end{bmatrix}}^{L_e} z = 0 \tag{3.41}$$

where the notation H_e, L_e, and x_e has been introduced. This implies that the intersection in (3.39) can be expressed as

$$\cap_{M_i \in \omega} \mathcal{O}(M_i) = \{z \in \mathbb{R}^{n_z} | \exists x_e \in \mathbb{R}^{nn_x}; H_e x_e + L_e z = 0\} \tag{3.42}$$

which according to (3.10) is equivalent to

$$\cap_{M_i \in \omega} \mathcal{O}(M_i) = \{z \in \mathbb{R}^{n_z} | N_{H_e} L_e z = 0\} \tag{3.43}$$

The matrix expression $N_{H_e} L_e$ in (3.43) can be simplified as follows

$$N_{H_e} L_e = \begin{bmatrix} N_{H[M_1]} & 0 & \cdots & 0 \\ 0 & N_{H[M_2]} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & N_{H[M_n]} \end{bmatrix} \begin{bmatrix} L \\ L \\ \vdots \\ L \end{bmatrix} = \begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \\ \vdots \\ N_{H[M_n]} \end{bmatrix} L \tag{3.44}$$

Note that $N_{H[M_i]}$ is zero-padded and therefore multiplied with L instead of $L[M_i]$. From (3.43) and (3.44) we obtain an alternative expression

$$\cap_{M_i \in \omega} \mathcal{O}(M_i) = \{z \in \mathbb{R}^{n_z} | \begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \\ \vdots \\ N_{H[M_n]} \end{bmatrix} L z = 0\} \tag{3.45}$$

for the intersection. Note that this is the representation that is used in the discussion concerning (3.37). By using the representation (3.45), it will be natural to extend the redundancy concept to family of equation sets next.

3.4.4 Redundancy of a Family of Equation Sets

As said before, the redundancy of a set M is equal to the maximum number of linearly independent consistency relations that can be derived from M. For any family ω of equations sets, it can be seen in (3.45) that a number of consistency relations can be related to ω. This motivates that the notion of redundancy can be generalized to a family of equation set and this will be done next. To make the definition of redundancy of a family ω as similar as possible to the redundancy definition of a set of equations, the representation (3.41) will be used in the definition instead of (3.45).

Definition 3.5 (Redundancy of a Family of Models). *Given a set M of linear equations (3.5) and a family of subsets $\omega = \{M_1, M_2, \ldots M_n\}$ where $M_i \subseteq M$ for all $M_i \in \omega$, the* ***redundancy of*** *ω is*

$$\varphi(\omega) = \text{rank}([H_e\, L_e]) - \text{rank}(H_e) \tag{3.46}$$

where H_e and L_e are defined as in (3.41).

Note that $\varphi(\{M\}) = \varphi(M)$, i.e. the definition of redundancy of a set of models is a generalization of redundancy of a model. The redundancy of a family of sets can be expressed in many different ways. For example, by using (3.14) we get

$$\varphi(\omega) = \text{rank}(N_{H_e} L_e) \tag{3.47}$$

or by using (3.45) a simplified redundancy expression is

$$\varphi(\omega) = \text{rank}\,(\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \\ \vdots \\ N_{H[M_n]} \end{bmatrix} L) \tag{3.48}$$

Compare this redundancy expression with the matrices in (3.37). It is then clear that, e.g. any pair of MO sets in (3.37) has redundancy 2. Furthermore, the redundancy expressed as in (3.48) shows that the generalized redundancy can, like the redundancy for models, be interpreted as the maximum number of linearly independent consistency relations that can be obtained.

3.4.5 Determine Consistency of Models by Testing MO Subsets

The conclusion of the example in (3.37) can in terms of the generalized redundancy be stated as follows. If a family of subsets of M has the same redundancy as M, then the consistency of M can be determined by checking the consistency of the subsets.

The next theorem shows that this principle is generally valid.

Theorem 3.6. *Given a set M of linear equations (3.5) and a family of subsets $\omega = \{M_1, M_2, \ldots M_n\}$ where $M_i \subseteq M$ for all $M_i \in \omega$, it holds that*

$$\mathcal{O}(M) = \cap_{M_i \in \omega} \mathcal{O}(M_i) \tag{3.49}$$

if and only if

$$\varphi(\omega) = \varphi(M) \tag{3.50}$$

To prove the theorem we introduce a notation for row-span as follows. If M is the rows of H then let the row-span of H be denoted by

$$\text{sp}(H) = \{\sum_{i \in M} \gamma_i H[\{i\}] | \gamma_i \in \mathbb{R}\} \tag{3.51}$$

Proof. Equation (3.49) holds if and only if

$$\text{sp}(\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \\ \vdots \\ N_{H[M_n]} \end{bmatrix} L) = \text{sp}(N_H L) \tag{3.52}$$

according to (3.45). By definition of null-space, it follows that $\text{sp}(N_{H[M_i]}) \subseteq \text{sp}(N_H)$ and then also that

$$\text{sp}(\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \\ \vdots \\ N_{H[M_n]} \end{bmatrix} L) \subseteq \text{sp}(N_H L) \tag{3.53}$$

This implies that we get equality if and only if (3.50) is true and the theorem follows. □

The next corollary shows that the L matrix need not be considered when determining the redundancy of a family of a PO sets.

Corollary 3.2. *Let M be a PO set and let $M_i \subset M$ be PO sets for all $i \in I = \{1, 2, \ldots, n\}$. Let the equations M be expressed by the matrices H and L. Then it holds that*

$$\mathcal{O}(M) = \cap_{i \in I} \mathcal{O}(M_i) \tag{3.54}$$

if and only if

$$\text{rank}\,(\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \\ \vdots \\ N_{H[M_n]} \end{bmatrix}) = \text{rank}\,(N_H) \tag{3.55}$$

Proof. Equality (3.54) holds according to Theorem 3.6 if and only if (3.50) holds. Since M is a PO set, it follows from Theorem 3.5 that $[H\,L]$ has full row-rank. This implies that

$$\text{rank}\,(\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \\ \vdots \\ N_{H[M_n]} \end{bmatrix}) = \text{rank}\,(\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \\ \vdots \\ N_{H[M_n]} \end{bmatrix} \begin{bmatrix} H & L \end{bmatrix}) = \text{rank}\,(\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \\ \vdots \\ N_{H[M_n]} \end{bmatrix} L) = \varphi\,(\omega) \tag{3.56}$$

and

$$\text{rank}\,(N_H) = \text{rank}\,(N_H[H\,L]) = \text{rank}\,(N_H L) = \varphi\,(M) \tag{3.57}$$

If these two rank equalities are used in (3.55), the corollary follows from Theorem 3.6. □

Note that the corollary basically shows that under the assumption that $[H\,L]$ has full row rank the redundancies in (3.50) can be simplified to be

$$\varphi\,(\omega) = \text{rank}\,(\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \\ \vdots \\ N_{H[M_n]} \end{bmatrix}) \tag{3.58}$$

and

$$\varphi\,(M) = \text{rank}\,(N_H) \tag{3.59}$$

Now, an example will illustrate how this theorem and corollary can be used.

Example 3.7

Consider the PO sets in (3.37). We first note that all sets in (3.37) are PO sets and that the MO sets are subsets of the PO set $\{3, 4, 5, 7\}$ with redundancy 2. By combining the rows of two MO sets for example $M_1 = \{3, 5, 7\}$ and $M_2 = \{4, 5\}$ we get the corresponding matrix

$$\varphi\,(\{M_1, M_2\}) = \text{rank}\,(\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \end{bmatrix} L) = \text{rank}\,(\begin{bmatrix} 0 & -1 & 0 & R_2 \\ 1 & -1 & 0 & 0 \end{bmatrix}) = 2 = \varphi\,(\{3, 4, 5, 7\}) \tag{3.60}$$

Theorem 3.6 then implies that

$$\mathcal{O}(\{3, 4, 5, 7\}) = \mathcal{O}(\{3, 5, 7\}) \cap \mathcal{O}(\{4, 5\}) \tag{3.61}$$

Hence the consistency of {3, 4, 5, 7} can be determined by the consistency of MO sets. Since all these sets are PO sets it is also possible to use Corollary 3.2. The corresponding rank test becomes

$$\text{rank}\,(\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \end{bmatrix}) = \text{rank}\,(\begin{bmatrix} 0 & 0 & 1 & 0 & -1 & 0 & R_2 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \end{bmatrix}) = 2 = \varphi\,(\{3,4,5,7\}) \quad (3.62)$$

An advantage with this rank condition is that by only using that the sets {3, 5, 7} and {4, 5} are PO sets, Corollary 3.1 implies that

$$\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \end{bmatrix} = \begin{bmatrix} 0 & 0 & n_1 & 0 & n_2 & 0 & n_3 \\ 0 & 0 & 0 & n_4 & n_5 & 0 & 0 \end{bmatrix} \quad (3.63)$$

where all $n_i \neq 0$. Without using H or L and from (3.63) it then follows that

$$\text{rank}\,(\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \end{bmatrix}) = \text{rank}\,(\begin{bmatrix} 0 & 0 & n_1 & 0 & n_2 & 0 & n_3 \\ 0 & 0 & 0 & n_4 & n_5 & 0 & 0 \end{bmatrix}) = 2$$

This means that without knowing the exact matrices H and L, it is sufficient to know that the sets {3, 5, 7} and {4, 5} are MO sets and subsets of the PO set {3, 4, 5, 7} with redundancy 2 to conclude that (3.61) holds.

In the next theorem this result will be shown to hold generally and the idea of proof is to use Corollary 3.1 in combination with Corollary 3.2 as in the example. Before we state the theorem we consider another example, investigating if the consistency of the no-fault model {1, 2, . . . , 7} can be determined by checking the consistency of MO sets. If this is possible, the number of needed MO sets must according to Corollary 3.2 be 3, because the redundancy of the no-fault model is 3. One way to do this is to chose three MO sets and test if (3.55) is true for these sets. One example is {4, 5}, {1, 2, 3, 4, 6}, and {3, 4, 7}. The next theorem shows that any PO set can be checked by MO subsets.

Theorem 3.7. *If M is a PO set with redundancy φ, then it follows that there exists a number of φ MO sets $M_i \subseteq M$ such that*

$$O(M) = \cap_i O(M_i) \quad (3.64)$$

The proof is constructive in the sense that it proposes a method to compute a family of φ MO sets that can be used to determine the consistency of the PO set.

Example 3.8

To give an example of the construction consider the no-fault model of the electrical circuit. It holds that $E = \{1, 2, 3, 4\}$ are linearly independent and rank $(H[M'])$ = rank (H). The remaining equations are {5, 6, 7}. By adding one equation to E the redundancy will be one and an MO set must therefore be included in the obtained set. The MO sets in $E \cup \{5\}$, $E \cup \{6\}$, and $E \cup \{7\}$ are $M_1 = \{4, 5\}$, $M_2 = \{1, 2, 3, 4, 6\}$, and $M_3 = \{3, 4, 7\}$ respectively. By using Corollary 3.1, it follows that

$$\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \\ N_{H[M_3]} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & n_1 & n_2 & 0 & 0 \\ n_3 & n_4 & n_5 & n_6 & 0 & n_7 & 0 \\ 0 & 0 & n_8 & n_9 & 0 & 0 & n_{10} \end{bmatrix} \quad (3.65)$$

where $n_i \neq 0$. Note that by construction, the last three columns constitute a diagonal matrix, i.e. the matrix (3.65) has full row-rank. Then Corollary 3.2 implies that these three MO sets can be used to determine the consistency of the no-fault model. By starting with three other linearly independent rows other triples of MO sets will be found.

Now, the proof of Theorem 3.7 follows.

Proof. The redundancy for all $M' \subseteq M$ is equal to

$$\varphi\,(M') = |M'| - \text{rank}\,(H[M']) \tag{3.66}$$

Let $E \subseteq M$ be rank (H) independent rows of H. No subset of E is a PO set, because $H[E]$ has full row-rank. If M' is set to M in (3.66) and it is used that $\varphi\,(M) = \varphi$ and rank $(H[M]) = |E|$, we get

$$\varphi\ = |M| - |E|$$

This implies that the set $M \backslash E$ contains φ equations and we will show that to each of these equations an MO set can be constructed. Let $e_i \in M \setminus E$ be arbitrarily chosen. Since $H[E]$ is a row basis of H, it follows that rank $(H[E \cup \{e_i\}])$ = rank $(H[E])$ and also that $\varphi\,(E \cup \{e_i\}) = 1$. Then there exists a PO set $M_i \subseteq E \cup \{e_i\}$ according to Theorem 3.2. This PO set has redundancy 1 and is therefore an MO set according to Lemma 3.2. Since $H[E]$ has full row rank, it follows that $e_i \in M_i$. If this and Corollary 3.1 is used for all $e_i \in M \setminus E$, then it follows that

$$\begin{bmatrix} N_{H[M_1]} \\ \vdots \\ N_{H[M_\varphi]} \end{bmatrix} = \begin{bmatrix} A & D \end{bmatrix} \tag{3.67}$$

where $A \in \mathbb{R}^{\varphi \times (|M|-\varphi)}$ and D is a diagonal matrix of dimension φ with non-zero diagonal elements. This matrix has obviously rank φ and the theorem follows from Corollary 3.2. □

Finally, we show that the consistency of any behavioral model can be determine by checking the consistency of all MO set.

Corollary 3.3. *Given any linear static behavioral model M_b, it follows that*

$$\mathcal{O}(M_b) = \cap_{M_i \in \mathcal{M}_{MO}} \mathcal{O}(M_i) \tag{3.68}$$

where $\mathcal{M}_{MO}$ is the family of all MO sets $M \subseteq M_b$.

Proof. According to Theorem 3.2 there exists a PO set $E \subseteq M_b$ such that

$$\mathcal{O}(M_b) = \mathcal{O}(E)$$

Theorem 3.7 then implies that

$$\mathcal{O}(M_b) = \mathcal{O}(E) = \cap_{E' \in \mathcal{M}'_{MO}} \mathcal{O}(E')$$

where $\mathcal{M}'_{MO}$ are all MO sets that is a subset of E. Hence $\mathcal{M}'_{MO} \subseteq \mathcal{M}_{MO}$ and the corollary follows. □

3.5 Selecting MO Sets

In the previous section, it was shown that it is sufficient to check the consistency of all MO sets in a behavioral model M_b, but according to Theorem 3.6 this is not necessary. The question is then how to find a family ω of MO sets such that (3.50) in Theorem 3.6 is fulfilled.

The consistency of a set with redundancy φ can always be determined by checking φ number of MO sets according to Theorem 3.7. This is possible if and only if the family of φ MO sets has redundancy φ according to Theorem 3.6. Not all families of φ MO subsets of the PO set have redundancy φ and this will be shown by the next example.

Example 3.9
The no-fault model $M_{\mathbf{NF}}$ for the electrical circuit example in (2.44) has redundancy $\varphi\left(M_{\mathbf{NF}}\right) = 3$, but for example the family $\omega = \{\{4,5\},\{3,5,7\},\{3,4,7\}\}$ of MO sets has redundancy 2 according to the discussion in Section (3.4.2). This means according to Theorem 3.6 that ω cannot be used to determine the consistency of $M_{\mathbf{NF}}$.

One alternative to find a possible triple of MO sets is of course to chose triplets and test them using (3.55) until one is found that satisfies this condition. However, the next theorems will provide guidelines to find a family ω of MO sets with the desired redundancy.

Theorem 3.8. *For any set of models ω it holds that*

$$\varphi\left(\omega\right) \leq \varphi\left(\cup_{M_i \in \omega} M_i\right) \tag{3.69}$$

Proof. Starting with (3.40) and substitute (3.10) and (3.45), we get

$$\{z \in \mathbb{R}^{n_z} | N_{H[\cup_{M_i \in \omega} M_i]} L z = 0\} \subseteq \{z \in \mathbb{R}^{n_z} | \begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \\ \vdots \\ N_{H[M_n]} \end{bmatrix} L z = 0\}$$

This means that

$$\text{rank}\ (\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \\ \vdots \\ N_{H[M_n]} \end{bmatrix} L) \leq \text{rank}\ (N_{H[\cup_{M_i \in \omega} M_i]} L)$$

By replacing the rank expressions by redundancy expressions according to (3.14) and (3.48), the inequality (3.69) is obtained. □

Theorem 3.9. *If ω is a set of PO sets and M is a PO set such that*

$$\cup_{M_i \in \omega} M_i \subset \cup_{M_i \in \omega} M_i \cup M \tag{3.70}$$

then it follows that

$$\varphi\left(\omega \cup \{M\}\right) \geq \varphi\left(\omega\right) + 1 \tag{3.71}$$

Proof. Let $M' = M \setminus \cup_{M_i \in \omega} M_i$. Since M is a PO set it follows that

$$N_{H[M]}[:, M'] \neq 0 \tag{3.72}$$

according to Corollary 3.1. Moreover $M_i \cap M' = \emptyset$ and this means that

$$N_{H[M_i]}[:, M'] = 0 \tag{3.73}$$

for all $M_i \in \omega$. From (3.72) and (3.73), it follows that $N_{H[M]}$ is not a linear combination of $N_{H[M_i]}$ for all $M_i \in \omega$. This and (3.58) imply

$$\varphi(\omega) = \text{rank}\,(\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \\ \vdots \\ N_{H[M_n]} \end{bmatrix}) < \text{rank}\,(\begin{bmatrix} N_{H[M_1]} \\ N_{H[M_2]} \\ \vdots \\ N_{H[M_n]} \\ N_{H[M]} \end{bmatrix}) = \varphi(\omega \cup \{M\})$$

which was to be proven. □

Example 3.10
To illustrate result Theorem 3.8 consider the previous example. There we found that $\omega = \{\{4,5\},\{3,5,7\},\{3,4,7\}\}$ could not be used to determine the consistency of $M_{\mathbf{NF}}$. If $\omega = \{\{4,5\},\{3,5,7\},\{3,4,7\}\}$ in (3.69), we get

$$\varphi(\omega) \leq \varphi(\{3,4,5,7\}) = 2 \tag{3.74}$$

which also confirms that ω cannot be used. From the definition of PO set and (3.69), it follows that a necessary condition for $\varphi(\omega) = 3$ is that the union of the sets in ω must contain all equations in $M_{\mathbf{NF}}$. Hence to find three MO sets with redundancy 3, it is necessary that all equations in $M_{\mathbf{NF}}$ are included in some of these MO sets.

Generally, assume that we want to find a family ω of MO subsets to determine the consistency of the PO set M. A necessary condition of ω is then according to (3.69) that

$$\cup_{M_i \in \omega} M_i = M$$

The result of Theorem 3.9 implies that a sufficient condition for finding a family $\omega = \{M_1, M_2, \ldots, M_n\}$ of MO sets with redundancy equal to the cardinality is that there exists a ordering $(a_1, \ldots, a_i, \ldots, a_n)$ of $\{1, 2, \ldots, n\}$ such that

$$\cup_{i=1}^{j} M_{a_i} \subset \cup_{i=1}^{j+1} M_{a_i} \tag{3.75}$$

for all $j \in \{1, 2, \ldots, n-1\}$.

Example 3.11
To give an example consider again the no-fault model for the electrical circuit example with the PO sets given in Figure 2.2. A family with 3 MO sets and with redundancy 3 is to be found. A sequence of three MO sets can be chosen as $\{4,5\}$, $\{3,5,7\}$, and finally $\{1,2,5,6,7\}$. These defines a sequence of strictly increasing sets, i.e.

$$\{4,5\} \subset \{4,5\} \cup \{3,5,7\} \subset \{4,5\} \cup \{3,5,7\} \cup \{1,2,5,6,7\} = M_{\mathbf{NF}} \tag{3.76}$$

and it follows that these three MO sets can be used the determine the consistency of the no-fault model. This shows how to find which MO sets to include in ω without using numerical computations once the MO sets are computed. Note that there are many other possible MO sets that can be used.

Another example is to use $\omega' = \{\{1,2,3,4,6\},\{3,5,7\},\{4,5\}\}$. To verify that $\varphi(\omega') = 3$, by using Theorem 3.9 an order of the MO sets is needed. The order they are presented in ω' can not be used since

$$\{1,2,3,4,6\} \cup \{3,5,7\} = M_{\mathbf{NF}} \tag{3.77}$$

However if the reversed order is used the condition in Theorem 3.9 is satisfied.

3.6 Linear Models with Constants

In this section, the theory given in this chapter is extended to behavioral models M_b of the type

$$H_b x + L_b z + c = 0 \tag{3.78}$$

where c is a constant vector. This type of models will naturally appear as the result of linearizations of non-linear models. It is assumed that there exists a value of x and z that satisfies (3.78), i.e.

$$\operatorname{rank}\left(\begin{bmatrix} H_b & L_b & c \end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix} H_b & L_b \end{bmatrix}\right) \tag{3.79}$$

Let

$$\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} x' \\ z' \end{bmatrix} + \alpha \tag{3.80}$$

define a variable translation where α is a constant vector to be determined. If this translation is applied to (3.78), then we get

$$\begin{bmatrix} H_b & L_b \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + c = \begin{bmatrix} H_b & L_b \end{bmatrix} \begin{bmatrix} x' \\ z' \end{bmatrix} + \begin{bmatrix} H_b & L_b \end{bmatrix} \alpha + c = 0 \tag{3.81}$$

From (3.79), it follows that there exists an $\alpha = \alpha_0$ such that

$$\begin{bmatrix} H_b & L_b \end{bmatrix} \alpha_0 = -c \tag{3.82}$$

By using $\alpha = \alpha_0$ in the variable translation the behavioral model in (3.78) is written

$$\begin{bmatrix} H_b & L_b \end{bmatrix} \begin{bmatrix} x' \\ z' \end{bmatrix} = 0 \tag{3.83}$$

i.e. in the same form as (3.2). Note that the matrices H_b and L_b are unchanged by the translation. Hence all theory developed for models of the type (3.2) is also valid for models of the type (3.78) as long as (3.79) holds true.

3.7 Redundancy and the Number of MO Sets

We have suggested to check the consistency of MO sets and then the number of MO sets will characterize the number of potential tests. In this section, it is shown that for a fixed order of redundancy φ, the number of MO sets is polynomial in the number of equations. Furthermore, it is shown that the redundancy is limited by the number of available sensors, which are often expensive, and therefore the redundancy is low in many applications.

3.7.1 Redundancy Dependence of the Number of Sensors

The redundancy depends on the number of available sensors n_s as follows. Let M_b be a set of equations partitioned into sensor equations M_1 and other equations M_2. Furthermore, let the vector of known variables be divided into two parts $z^T = [\mathbf{y}^T\,\mathbf{u}^T]$ where $\mathbf{y}$ is the measured signals corresponding to the sensors and $\mathbf{u}$ is the control signals. The equations M_b partitioned into M_1 and M_2 can then be written as:

$$\overbrace{\begin{bmatrix} H_1 \\ H_2 \end{bmatrix}}^{H} x + \overbrace{\begin{bmatrix} I & 0 \\ 0 & L_2 \end{bmatrix}}^{L} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \end{bmatrix} = 0 \tag{3.84}$$

Note that the cardinality of M_1 is equal to the number of sensors n_s. A model does normally not specify any algebraic dependence between control signals, i.e. rank $(N_{H_2} L_2) = 0$ or equivalently $\varphi\,(M_2) = 0$. Using this type of model, the redundancy is less or equal to the number of sensors, i.e.

$$\varphi\,(M_b) \leq n_s \tag{3.85}$$

and it can be shown as follows. By using that

$$\text{rank}\,([H\,L]) = |M_1| + \text{rank}\,([H[M_2]\,L[M_2]])$$

and

$$\text{rank}\,(H) \geq \text{rank}\,(H[M_2]) \tag{3.86}$$

an upper limit of the redundancy can be found as

$$\begin{aligned} \varphi\,(M_b) &= \text{rank}\,([H\,L]) - \text{rank}\,(H) \\ &\leq |M_1| + \text{rank}\,([H[M_2]\,L[M_2]]) - \text{rank}\,(H[M_2]) \\ &= n_s + \varphi\,(M_2) = n_s \end{aligned}$$

Furthermore equality in (3.85) holds if and only if equality in (3.86) holds. This is especially true for example when $H[M_2]$ is invertible, i.e. has full column rank. In summary, the number of sensors limits the redundancy of the model. To give an example, the electrical circuit has 3 sensors and the no-fault model has redundancy 3. Next we will study the how the number of MO sets depends on the redundancy.

3.7.2 The Number of MO Sets

For a fixed order of redundancy φ, the number of MO sets in M_b is polynomial in the number n_e of equations. This follows from the fact that in the worst case every subset of M_b, with one more equation than the number of unknowns is an MO set. This means that the number of MO sets is less or equal to the number of ways to choose $n_e - \varphi + 1$ equations from the original n_e equations, i.e.

$$\binom{n_e}{n_e - \varphi + 1} = \frac{n_e!}{(n_e - \varphi + 1)!(\varphi - 1)!} \propto n_e(n_e - 1)\cdots(n_e - \varphi + 2) \approx n_e^{\varphi - 1} \tag{3.87}$$

The number of such sets grows polynomially in the number of equations.

3.8 Diagnosing all Behavioral Models

So far in this chapter, we have only been concerned to fulfill (2.34) for one single behavioral mode. In this section, we extend the discussion to the case when all behavioral modes in $\mathbf{B}$ are considered.

3.8.1 Sufficient to Test All MO Sets

In the next theorem it is stated that given a diagnosis model, it is sufficient to check the consistency of all feasible MO sets to determine the consistency of all behavioral models.

Theorem 3.10. *If $\mathbb{M}$ is a diagnosis model with a set $\boldsymbol{M} = \cup_{b \in \mathbf{B}} M_b$ of linear static equations such that M_b is on the form* (3.78) *and satisfies* (3.79) *for all $b \in \mathbf{B}$, then the set ω of all feasible MO sets in $\boldsymbol{M}$ satisfies* (2.34) *for all $b \in \mathbf{B}$.*

Proof. Let b be an arbitrary behavioral mode $b \in \mathbf{B}$. First, since M_b fulfills (3.79) the model can be rewritten as (3.83). Second, any MO sets $M \subseteq M_b$ is a feasible MO set, i.e. $M \in \omega$. These two facts imply that (2.34) is satisfied for b according to Corollary 3.3. Since b was arbitrarily chosen the theorem follows. □

A feasible MO set M normally satisfy assump$(M) \subset \mathbf{B}$. However if assump(M) $= \mathbf{B}$ it means that M cannot be inconsistent and need not be included in ω. Let $\omega = \{M_i | i \in I\}$ be the set of all feasible MO sets in a diagnosis model $\mathbb{M}$. For each feasible MO sets in ω, i.e. for each $i \in I$, a test δ_i is constructed such that Φ_i = assump(M_i), $T_i(z) = N_{H[M_i]} L z$, and $\mathcal{R}_i^C = \{0\}$. Then Theorem 2.1, Theorem 2.2, and Theorem 3.10 imply that the diagnosis system $\Delta = \{\delta_i | i \in I\}$ is sound and complete with respect to the diagnosis model $\mathbb{M}$.

Example 3.12
Consider the electrical circuit example and the Hasse diagram in Figure 2.2 showing all PO sets contained in its diagnosis model. All MO sets are in this example feasible models because $\boldsymbol{M} = M_{\mathbf{NF}}$. A consequence of Theorem 3.10 is that there exists a sound diagnosis system based on the 8 MO sets. The matrices $N_{H[M_i]} L$ for all MO sets $M_i \in \boldsymbol{M}$ are

$$
\begin{array}{llrrrrl}
MO & & U & y_V & y_I & y_{I2} & \\
\hline
\{1,2,3,4,6\} & [& -(R_1+R_2) & 0 & R_1 R_2 & 0 &] \\
\{1,2,3,5,6\} & [& 0 & -(R_1+R_2) & R_1 R_2 & 0 &] \\
\{1,2,3,6,7\} & [& 0 & 0 & R_1 & -(R_1+R_2) &] \\
\{1,2,4,6,7\} & [& -1 & 0 & R_1 & -R_1 &] \\
\{1,2,5,6,7\} & [& 0 & -1 & R_1 & -R_1 &] \\
\{3,4,7\} & [& -1 & 0 & 0 & R_2 &] \\
\{3,5,7\} & [& 0 & -1 & 0 & R_2 &] \\
\{4,5\} & [& 1 & -1 & 0 & 0 &]
\end{array}
\tag{3.88}
$$

and these are used to form the test quantities $T_i(z) = N_{H[M_i]} L z$. At this point, it might be interesting to reread Section 2.6 in the light of the new theory developed in this chapter and note especially that by checking the consistency of the 8 MO sets the consistency of all $2^6 = 64$ behavioral models can be determined.

3.9 Minimizing The Number of Tests

In the previous section, it was shown that it is sufficient to check all feasible MO sets in a diagnosis model. However, it has been shown that it is not necessary to check all MO sets in general, i.e. some MO sets do not need to be tested. This raises the question of how to find a smallest family ω of MO sets such that (2.34) is fulfilled for all $b \in \mathbf{B}$. In this section an algorithm for finding not only one but all minimal solutions ω will be presented and some examples will be discussed.

3.9.1 Algorithm

In this section an algorithm, Algorithm 2, will be presented that can be used to find all minimum cardinality solutions ω to (2.34) given a diagnosis model. The main purpose of the algorithm is to explain how minimal cardinality solutions can be computed in a straightforward way. The focus is not to optimize the computational complexity.

The solutions have the following property that will be used in the algorithm. If ω satisfies (2.34) for all behavioral modes $b \in \mathbf{B}$, then any superset $\hat{\omega} \supset \omega$ will also satisfy (2.34) for all behavioral modes $b \in \mathbf{B}$. This means that there exist minimal families ω of MO sets that satisfy (2.34). A minimum cardinality solution ω to (2.34) must therefore be among the minimal families that solves (2.34).

Instead of using that (2.34) holds for all $b \in \mathbf{B}$ an equivalent condition according to Theorem 3.2 is that

$$\mathcal{O}(M') = \bigcap_{M \in \omega} \mathcal{O}(M) \tag{3.89}$$

for all PO sets M' such that $\mathcal{O}(M') = \mathcal{O}(M_b)$ for some $b \in \mathbf{B}$ will be used. If $\mathbf{B}$ is the set of all single-fault behavioral modes and the no-fault mode, then the PO sets M' in Figure 2.2 are the PO sets with some behavioral mode in $\mathbf{B}$.

The algorithm takes as inputs the set $\mathcal{M}_{MO}$ of all MO sets, all PO sets with a behavioral mode in $\mathbf{B}$, and the model equations $\boldsymbol{M}$ and finds the set Ω of all minimal solutions ω.

Let $P = (P_1, \ldots P_n)$ be an ordered set of all PO sets with a behavioral mode in $\mathbf{B}$. Let Ω_i be the set of all minimal solutions ω such that $(P_1, \ldots P_i)$ are tested. The basic idea is to start with $\Omega_0 = \emptyset$ and then extend the solutions in Ω_{i-1} to form the set Ω_i until we obtain the set Ω_n which is equal to the output set Ω.

The set Ω_i is computed using Ω_{i-1} and the i:th PO set P_i. This computation consists of two steps. First, the set Ω'_i of all minimal families of MO sets ω' that test P_i is computed in Algorithm 1. Second, Ω'_i and Ω_{i-1} are merged to form the set Ω_i. These steps will now be explained.

Given a PO set P_i and a family of subsets $\omega' = \{M_1, M_2, \ldots M_n\}$ where $M_j \subseteq P_i$ for all $M_j \in \omega'$, it holds that

$$\mathcal{O}(P_i) = \bigcap_{M \in \omega'} \mathcal{O}(M) \tag{3.90}$$

if and only if

$$\varphi(\omega') = \varphi(P_i) \tag{3.91}$$

according to Theorem 3.6. Hence $\omega' \in \Omega'_i$ if and only if $\omega' = \{M_1, M_2, \dots M_n\}$ is a minimal set that satisfies (3.91) and where $M_j \subseteq P_i$ for all $M_j \in \omega'$.

Since redundancy of a set can be expressed as (3.48), i.e. the rank of a matrix where each row corresponds to an MO set, it follows that any minimal set ω' that satisfies (3.91) must contain exactly $\varphi(P_i)$ MO sets. This is an important property that can be used when generating all such sets and the computation can be done as follows.

Algorithm 1. Ω'_i = `MinimalTestSets`$(P_i, \mathcal{M}_{MO})$

```
$\Omega'_i = \emptyset$;
$\alpha := \{M \in \mathcal{M}_{MO} | M \subseteq P_i\}$;
$\hat{\Omega} := \{\omega' \subseteq \alpha \,|\, |\omega'| = \varphi(P_i)\}$;
for each $\omega' \in \hat{\Omega}$ do

    if $\varphi(\omega') = \varphi(P_i)$ do

        Insert $\omega'$ in $\Omega'_i$;

    end if

end for
return $\Omega'_i$
```

Given a PO set P_i and the set $\mathcal{M}_{MO}$ of all MO models, the output set Ω'_i in Algorithm 1 contains all minimal solutions $\omega' \subseteq \mathcal{M}_{MO}$ of (3.90). As said before, we will consider P_i to be a PO set and $\mathcal{M}_{MO}$ to be a set of all MO sets. However, the algorithm is not restricted to these inputs. In general, P_i can be any set of linear static equations and $\mathcal{M}_{MO}$ can be any set of linear static models.

Example 3.13

To illustrate how `MinimalTestSets` works consider the electrical circuit model (2.44) with PO sets given in Figure 2.2. Assume that $P_1 = \{1, 2, 3, 4, 5, 6\}$ and that the MO sets are enumerated from left to right in Figure 2.2 and let the i:th MO set be denoted by M_i. The set α defined in Algorithm 1 is for this example

$$\alpha = \{M_1, M_2, M_8\}$$

Since $\varphi(\{1, 2, 3, 4, 5, 6\}) = 2$ the set $\hat{\Omega}$ is

$$\hat{\Omega} = \{\{M_1, M_2\}, \{M_1, M_8\}, \{M_2, M_8\}\}$$

Each set $\omega' \in \hat{\Omega}$ satisfies (3.91) and the output set Ω'_i is $\Omega'_i = \hat{\Omega}$.

If the set of all minimal solutions Ω'_i of the i:th PO set in P_i is computed using Algorithm 1 and if the set of all minimal solutions for all the $i-1$ first PO sets in P is Ω_{i-1}, we will next explain how to compute the extended solution Ω_i. This will first be illustrated by continuing Example 3.13.

Example 3.14

Assume that **B** is the set of all single-faults and the no-fault mode. Assume that the PO sets corresponding to behavioral modes in **B** are enumerated from left to right and then by increasing redundancy. This order defines the list P. The first PO set P_1 in P is $P_1 = \{1, 2, 3, 4, 5, 6\}$ for behavioral mode **S3** and the set of all minimal families ω of MO sets that satisfy (2.34) for **S3** is according to Example 3.13 equal to

$$\Omega_1 = \{\{M_1, M_2\}, \{M_1, M_8\}, \{M_2, M_8\}\}$$

The next PO set in P is $P_2 = \{1, 2, 3, 4, 6, 7\}$, and the set Ω'_2 obtained using Algorithm 1 of all minimal families of MO sets that satisfy (2.34) for **S1** is

$$\Omega'_2 = \{\{M_1, M_2\}, \{M_1, M_3\}, \{M_1, M_6\}, \{M_2, M_3\}, \{M_2, M_6\}, \{M_3, M_6\}\}$$

Next, the solutions in Ω_1 and in Ω'_2 are combined to form a solution when considering both PO set P_1 and P_2. Since $\omega \in \Omega_1$ is a solution for P_1 and $\omega' \in \Omega'_2$ is a solution for P_2, it follows that $\omega \cup \omega'$ will be a solution, not necessarily minimal, to the problem of testing both P_1 and P_2. By taking the union of all pairs in the sets Ω_1 and Ω_2 and then remove non-minimal solutions, we obtain

$$\begin{aligned}\Omega_2 = \{&\{M_1, M_2, M_3\}, \{M_1, M_2, M_4\}, \{M_1, M_2, M_6\}, \{M_1, M_3, M_8\}, \{M_1, M_4, M_8\},\\ &\{M_1, M_6, M_8\}, \{M_2, M_3, M_4, M_8\}, \{M_2, M_3, M_6, M_8\}, \{M_2, M_4, M_6, M_8\}\}\end{aligned}$$

An example of a non-minimal solution that has been removed to obtain this set is $\{M_1, M_2, M_3, M_6\}$ which is produced by taking the union of $\{M_1, M_2\} \in \Omega_1$ and $\{M_3, M_6\} \in \Omega_2$. By construction, Ω_2 contains all minimal sets ω that satisfy (3.49) for both P_1 and P_2.

In the example we saw that Ω_i can be obtained from Ω_{i-1} and Ω'_i by first computing

$$\Omega_i = \{\omega' \cup \omega | \omega \in \Omega_{i-1}, \omega' \in \Omega'_i\} \tag{3.92}$$

and then pruning all non-minimal sets in Ω_i. Let these operations be denoted by `Extend` such that Ω_i can be expressed as

$$\Omega_i = \texttt{Extend}(\Omega_{i-1}, \Omega'_i) \tag{3.93}$$

Remember that $P = (P_1, \ldots P_n)$ is an ordered set of all PO sets with a behavioral mode in **B**. The output is invariant under permutations, but the computational complexity is not. The computational complexity is closely related to the size of Ω_i in all steps. A large redundancy of a PO set P_i leads in general to a large set Ω'_i. To reduce the size of the initial sets Ω_i a heuristic is then to order the PO sets in increasing redundancy.

Now, the main algorithm can now be summarized as follows.

Algorithm 2. $\Omega = \texttt{AllMinimalTestSets}(P, \mathcal{M}_{MO})$

```
$\Omega = \emptyset$;
for $i = 1$ to $n$ do

    $\Omega' := \texttt{MinimalTestSets}(P_i, \mathcal{M}_{MO})$;

    $\Omega := \texttt{Extend}(\Omega, \Omega')$;

end for
return $\Omega$
```

Given a list P of PO sets and the set $\mathcal{M}_{MO}$ of all MO models, the output set Ω of Algorithm 2 contains all minimal solutions $\omega \subseteq \mathcal{M}_{MO}$ of (3.89).

The minimum number of tests can now be obtained by testing the MO sets in any minimum cardinality set $\omega \in \Omega$. Sometimes the set Ω is large and therefore computationally expensive to compute. Since the objective is to compute only one minimum cardinality solution it is not necessary to compute all minimal solutions. By the following small modification of the above described algorithm, it is possible to define an algorithm where the output set Ω will be the set of all minimum cardinality sets ω. If the minimum cardinality sets ω have cardinality m, then only the minimum cardinality sets ω is obtained in the output of Algorithm 2 if (3.92) is replaced by

$$\Omega_i = \{\omega' \cup \omega | \omega \in \Omega_{i-1}, \omega' \in \Omega_i', |\omega' \cup \omega| \leq m\} \tag{3.94}$$

The number m can be found as the minimum number that gives a non-empty output set Ω of the modified Algorithm 2 and this set is then the desired set.

Since all minimum cardinality sets can be computed additional objectives, such as computational aspects of the residuals, can be added to chose the best minimum cardinality set.

3.9.2 Examples

Now, both Theorem 3.10 and Algorithm 2 will be used to obtain MO sets ω for some variants of the electrical circuit example.

Example 3.15
First, consider again the electrical circuit when the set **B** contains all multiple fault behavioral modes and the no-fault mode. This means according to Figure 2.2 that the input list P in Algorithm 2 contains all PO sets and $\mathcal{M}_{MO}$ contains all MO sets in Figure 2.2. The output set Ω contains in this case only one minimal set $\omega = \mathcal{M}_{MO}$ including all MO sets in Figure 2. Therefore $\mathcal{M}_{MO}$ is the unique minimum cardinality solution. This means that the minimal number of tests required to obtain soundness is 8.

To explain the result, take for instance behavioral mode **R1&R2** with its corresponding behavioral model $\{1, 4, 5, 6, 7\}$. The MO set $\{4, 5\}$ is the only PO set with the same observation set as $\{1, 4, 5, 6, 7\}$. Hence to fulfill (2.34) for behavioral mode **R1&R2**, the MO set $\{4, 5\}$ has to be included in ω. This can be seen in Figure 2.2 where **R1&R2** is among the behavioral models listed below $\{4, 5\}$. In the figure, it can be seen that each MO set has an observation set equal to an observation set to some behavioral model. Therefore, all MO sets must in this case be included in ω.

Example 3.16
Assume this time that only single faults are to be considered, i.e. **B** = {**NF**, **R1**, **R2**, **B**, **S1**, **S2**, **S3**}. For soundness, it is required that the consistency of all behavioral models M_b such that $b \in$ **B** can be determined. By looking at the Hasse diagram in Figure 2.2, it can be seen that this is equivalent to determine that consistency of all PO sets with redundancy 2 and 3.

Since the model equations are the same as when considering all multiple faults, Theorem 3.10 again implies that all 8 MO sets are sufficient. Another possibility to satisfy (2.34) for all $b \in \mathbf{B}$ is to include the 5 PO sets with redundancy 2 in ω. However the redundancy is 2 for each PO set and this choice can therefore be seen as testing the consistency of $5 \times 2 = 10$ consistency relations compared to test the 8 MO sets.

Not all MO sets need to be tested in order to fulfill (2.34) for all behavioral modes in $\mathbf{B} = \{\mathbf{NF}, \mathbf{R1}, \mathbf{R2}, \mathbf{B}, \mathbf{S1}, \mathbf{S2}, \mathbf{S3}\}$. To compute the minimum number of MO sets needed Algorithm 2 is applied to the inputs P equal to the set of all PO sets with redundancy 2 and 3 and $\mathcal{M}_{MO}$ equal to the set of all MO sets in Figure 2.2. The output set Ω contains 13 minimal sets of which 12 are of minimum cardinality 5. One of these minimum cardinality sets is

$$\omega = \{\{1,2,3,4,6\},\{1,2,3,6,7\},\{1,2,5,6,7\},\{3,4,7\},\{4,5\}\}$$

Hence the minimum number of tests is 5.

In the following two examples, an extension to the electrical circuit model is considered.

Example 3.17

Next all multiple faults are again considered. In addition to the model given in (2.44), the battery is modeled to two behavioral modes $B \in \{\text{NF}, \text{F}\}$ where F stands for "faulty". A faulty battery is assumed to have no voltage across the battery, i.e. $V = 0$ if $B = \text{F}$. The extended model becomes

$$\begin{array}{lcrl}
\text{Assumption} & \text{Equation} & \text{Expression} & \\
\hline
 & (1) & I - I_1 - I_2 & = 0 \\
R1 = \text{NF} & (2) & V - I_1 R_1 & = 0 \\
R2 = \text{NF} & (3) & V - I_2 R_2 & = 0 \\
B = \text{NF} & (4) & V - U & = 0 \\
S1 = \text{NF} & (5) & V - y_V & = 0 \\
S2 = \text{NF} & (6) & I - y_I & = 0 \\
S3 = \text{NF} & (7) & I_2 - y_{I_2} & = 0 \\
B = \text{F} & (8) & V & = 0
\end{array} \tag{3.95}$$

The equations 4 and 8 are mutually exclusive. The set $\{4,8\}$ is therefore an example of an MO set that is not a feasible model. This MO set need not be considered according to Theorem 3.10. Since a subset of a feasible model is feasible, the set of all feasible models can be represented as all subsets of the maximal feasible models. The maximal feasible models in the example are $M \backslash \{8\} = M_{\mathbf{NF}}$ and $M \backslash \{4\} = M_{\mathbf{B}}$. The Hasse diagram for (3.95) is shown in Figure 3.1. This shows the subset relation of all feasible PO sets for the model. The two maximal feasible models are the ones on redundancy level 3.

The additional feasible MO sets found in the extended model (3.95) are $\{1,2,3,6,8\}, \{1,2,6,7,8\}, \{3,7,8\}, \{5,8\}$. If the input list P in Algorithm 2 contains all PO sets with behavioral modes in Figure 3.1 and $\mathcal{M}_{MO}$ contains all MO sets in Figure 3.1, then the output set Ω consists of one minimal set ω including the 8 MO sets in Figure 3.1 with behavioral modes. Hence it is necessary and sufficient to check the consistency of all 8 MO sets which includes either equation 4 or 8. The matrices $N_{H[M_i]}L$ for the MO sets $M_i \in \mathbf{M}$ are the ones

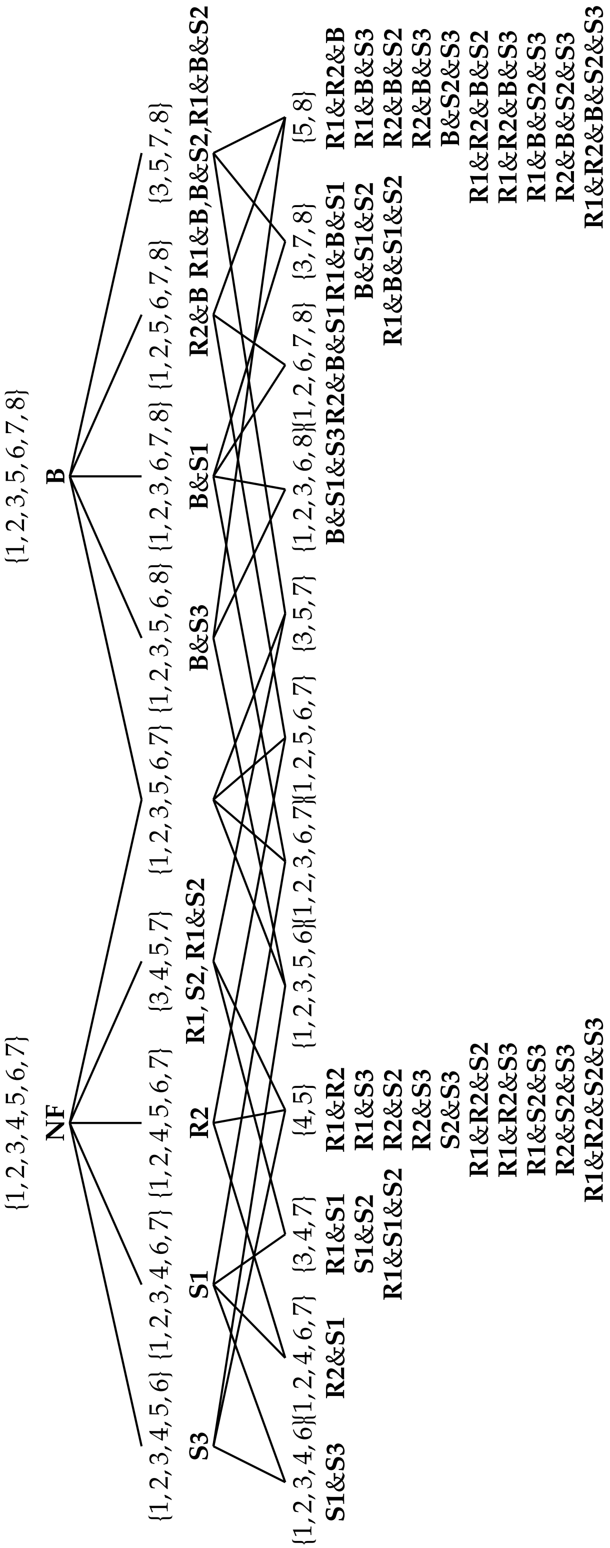

Figure 3.1: All feasible PO subsets of (3.95).

shown in (3.88) and

$$\begin{array}{lcccccc} MO & & U & y_V & y_I & y_{I2} & \\ \hline \{1,2,3,6,8\} & [& 0 & 0 & 1 & 0 &] \\ \{1,2,6,7,8\} & [& 0 & 0 & 1 & -1 &] \\ \{3,7,8\} & [& 0 & 0 & 0 & 1 &] \\ \{5,8\} & [& 0 & 1 & 0 & 0 &] \end{array} \tag{3.96}$$

Example 3.18
The last example is to consider the model in (3.95) but now with $\mathbf{B} = \{\mathbf{NF}, \mathbf{R1}, \mathbf{R2}, \mathbf{B}, \mathbf{S1}, \mathbf{S2}, \mathbf{S3}\}$. Let the input list P in Algorithm 2 contain all PO sets in Figure 3.1 with behavioral modes in $\mathbf{B} = \{\mathbf{NF}, \mathbf{R1}, \mathbf{R2}, \mathbf{B}, \mathbf{S1}, \mathbf{S2}, \mathbf{S3}\}$ and let $\mathcal{M}_{MO}$ contain all MO sets in Figure 3.1. The output set Ω contains 73 minimal sets of which 63 are of minimum cardinality 6. If the MO sets M_i are enumerated from left to right in Figure 3.1 one such set is

$$\omega = \{M_1, M_2, M_4, M_5, M_8, M_9\}$$

The number of minimal solutions ω is large if P includes few PO sets with low redundancy, as in the single fault examples, compared to the case when many MO sets are included in P, as in the multiple fault examples. Since the computational complexity of finding all minimum cardinality sets ω depends on the number of solutions ω, the same can be said about the computational complexity of computing Ω.

3.10 Fault Influence on Residuals

In the diagnosis system design approach used here, i.e. starting from the model as described in Section 2.3.3, the idea is to construct residuals for MO sets. Any test quantity based on the MO set M of equations $Hx + Lz + c = 0$ can be written with some basis N_H of the left null space of H as

$$r = N_H(Lz + c) \tag{3.97}$$

A scaling factor is the only design freedom for a residual based on the MO set M. Since scaling does not affect which faults that the residual r is influenced by, any residual based on the same MO set will be influenced by the same faults.

Fault influence on residuals is typically analyzed when faults are modeled with fault variables. In this case a residual r is said to be influenced by a fault F, if the residual r is sensitive to a change in the corresponding fault variable f, i.e. if $f \neq 0$ implies that $r \neq 0$ in the linear case. In the framework used here, faults need not be explicitly modeled with fault variables and the meaning of fault influence on a residual can not be defined in this way. A formulation of fault influence that generalizes to the situation without fault variables is the following. In the linear case, it is equivalent that a residual r is sensitive to a fault variable f corresponding to fault F and that $r \neq 0$ is consistent with the assumption that the system to be diagnosed is in fault mode sys $= F$. To make the next definition generally valid, we will use behavioral mode instead of fault mode.

Definition 3.6 (Influenced by a Behavioral Mode)**.** *A residual r is* ***influenced by a behavioral mode*** *b if*

$$\exists z \in \mathcal{O}(M_b) : r(z) \neq 0 \tag{3.98}$$

This means that a residual r is influenced by a fault mode F if $r \neq 0$ is consistent with the behavioral model M_F. This formulation of fault influence does not require that the fault F is explicitly modeled with a fault variable f. Hence this is what we mean with fault influence on a residual r in the general case.

In this section the fault influence on a residual r based on an MO set M is investigated and the main result is, under some general assumptions given later, that the r is influenced by the faults, or more general the behavioral modes, $b \in (\text{assump}(M))^C$. Before a general linear static model is discussed, we will consider a special case where all faults are explicitly modeled with fault variables.

3.10.1 Fault Influence in Models with Fault Variables

Assume that the no-fault behavior of a process can be described by the linear static model

$$H'\,x + L'\,z = 0 \tag{3.99}$$

Furthermore, assume that there are m single faults modeled by a vector f. For the variable f_i it holds that $f_i = 0$ if fault i is not present. The static linear model (2.26) would then be

$$H'x + L'z + F'f = 0 \tag{3.100}$$

This model can be written in the framework used here as follows. Each fault i can be related to a component c_i which can either be in the no-fault mode NF or the faulty mode F. The model then becomes

$$\begin{array}{cc} c_1 = NF & f_1 = 0 \\ \vdots & \vdots \\ c_m = NF & f_m = 0 \\ & H'x + L'z + F'f = 0 \end{array} \tag{3.101}$$

The variables in f are considered to be unknown variables. Assume that we pick out an MO set of equations in (3.101). These equations can, after a renumeration of the faults, be written as

$$\begin{array}{cc} c_1 = NF & f_1 = 0 \\ \vdots & \vdots \\ c_n = NF & f_n = 0 \\ & Hx + Lz + Ff = 0 \end{array} \tag{3.102}$$

where $n \leq m$. Let the fault vector f be partitioned such that the n first fault variables $f_1, \ldots, f_n$ form a vector f_a and the rest of the faults are included in f_b, i.e.

$$f = \begin{bmatrix} f_a \\ f_b \end{bmatrix} \tag{3.103}$$

Furthermore, let $F_a = F[:, \{1, \ldots, n\}]$ and $F_b = F[:, \{n+1, \ldots, m\}]$. The model equations (3.102) can now be written in the form (3.5) as

$$\underbrace{\begin{bmatrix} I & 0 & 0 \\ F_a & F_b & H \end{bmatrix}}_{\tilde{H}} \underbrace{\begin{bmatrix} f_a \\ f_b \\ x \end{bmatrix}}_{\tilde{x}} + \underbrace{\begin{bmatrix} 0 \\ L \end{bmatrix}}_{\tilde{L}} z = 0 \tag{3.104}$$

Any residual for this MO set can be written as

$$r = N_{\tilde{H}} \tilde{L} z \tag{3.105}$$

where the scaling α in (3.97) can be included in $N_{\tilde{H}}$. To find out the fault influence on the residual r, we want to express r only in the fault vector f.

To do this, we will first express the residual with the sub-matrices given in (3.104). Let the row-vector $N_{\tilde{H}}$ be partitioned into $[N_1\, N_2]$. Since $N_{\tilde{H}} \tilde{H} = 0$ we get that

$$\begin{bmatrix} N_1 & N_2 \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ F_a & F_b & H \end{bmatrix} = \begin{bmatrix} N_1 + N_2 F_a & N_2 F_b & N_2 H \end{bmatrix} = 0 \tag{3.106}$$

This implies that

$$N_2 = N_{[F_b\, H]} \tag{3.107}$$

and then also that

$$N_1 + N_{[F_b\, H]} F_a = 0 \tag{3.108}$$

From that $N_{\tilde{H}} = [N_1\, N_{[F_b\, H]}]$ and the definition of $\tilde{L}$ in (3.104), it follows that the residual in (3.105) can be expressed as

$$r = N_{\tilde{H}} \tilde{L} z = \begin{bmatrix} N_1 & N_{[F_b\, H]} \end{bmatrix} \begin{bmatrix} 0 \\ L \end{bmatrix} z = N_{[F_b\, H]} L z \tag{3.109}$$

Now, by multiplying the lower part of (3.104) with $N_{[F_b\, H]}$, we get

$$N_{[F_b\, H]} F_a f_a + N_{[F_b\, H]} F_b f_b + N_{[F_b\, H]} H x + N_{[F_b\, H]} L z = 0 \tag{3.110}$$

On the left-hand side the second and third term is zero and the last term is according to (3.109) equal to r. This means that the residual is equal to

$$r = -N_{[F_b\, H]} F_a f_a \tag{3.111}$$

Finally, by using (3.108) and that $N_1 = N_{\tilde{H}}[:, \{1, \ldots, n\}]$, the fault influence on the residual r is given by

$$r = N_{\tilde{H}}[:, \{1, \ldots, n\}] f_a \tag{3.112}$$

The residual is influenced by all faults in f_a where the corresponding column in the row-vector $N_{\tilde{H}}[:, \{1, \ldots, n\}]$ is non-zero. Since (3.102) is an MO set of equations, it follows from Corollary 3.1 that all columns in $N_{\tilde{H}}$ are non-zero and a consequence of this is that all columns in $N_{\tilde{H}}[:, \{1, \ldots, n\}]$ are non-zero. Hence the residual r in (3.105) is influenced by all faults in f_a and not influenced by any faults in f_b. An interpretation of (3.107) is that the faults f_b are decoupled and it will be shown that the resulting residual will not be influenced by any fault in f_b. Hence the partition of the faults in (3.103) can be interpreted as not decoupled faults f_a and decoupled faults f_b. The result of this discussion is summarized in the following theorem.

Theorem 3.11. *Given a linear static model where all faults are modeled explicitly as in* (3.101) *and an MO subset of equations* (3.102)*, any residual* $r = N_{\tilde{H}}\tilde{L}z$ *using the notation in* (3.104) *will have the fault influence*

$$r = \begin{bmatrix} N_{\tilde{H}}[:, \{1, \ldots, n\}] & 0 \end{bmatrix} \begin{bmatrix} f_a \\ f_b \end{bmatrix} \tag{3.113}$$

where $N_{\tilde{H}}[:, \{i\}] \neq 0$ *for all* $i \in \{1, \ldots, n\}$*.*

Proof. The theorem follows directly from the formal discussion starting with (3.101) and ending with Theorem 3.11. □

Usually, we will start with a diagnosis model with the set $\boldsymbol{M}$ of equations (3.101) and form a residual based on an MO set $M \subseteq \boldsymbol{M}$. If the set $\boldsymbol{M}$ of equations (3.101) is written on matrix form with matrices $\tilde{H}'$ and $\tilde{L}'$, then the result of Theorem 3.11 can then be reformulated in terms of matrices $\tilde{H}'$ and $\tilde{L}'$ as follows. Given any MO set $M \subseteq \boldsymbol{M}$, any residual based on M can the be written

$$r = N_{\tilde{H}'[M]}\tilde{L}'z \tag{3.114}$$

and its fault influence is given by

$$r = N_{\tilde{H}'[M]}[:, \{1, \ldots, m\}]f \tag{3.115}$$

according to Theorem 3.11.

Example 3.19
To exemplify the result of Theorem 3.11 consider the model

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} z + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} f = 0 \tag{3.116}$$

which written in the form (3.101) is

$$\begin{array}{l} c_1 = NF \\ c_2 = NF \\ c_3 = NF \\ \\ \end{array} \qquad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ x \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} z = 0 \tag{3.117}$$

where the matrices are called $\tilde{H}'$ and $\tilde{L}'$ respectively. This model includes one MO set, i.e. the set $\{1, 2, 4, 5\}$. By computing the vector

$$N_{\tilde{H}[\{1,2,4,5\}]} = \begin{bmatrix} -1 & 1 & 0 & 1 & -1 \end{bmatrix}$$

a residual and its fault influence is then given by

$$r = \begin{bmatrix} 1 & -1 \end{bmatrix} z = \begin{bmatrix} -1 & 1 & 0 \end{bmatrix} f \tag{3.118}$$

according to (3.114) and (3.115) respectively.

3.10.2 Fault Influence in General Linear Static Models

In the previous discussion all faults were explicitly included in the model as fault variables and then it was possible to quantify the influence of a fault on a residual. The next theorem generalizes the result in Theorem 3.11 to general linear static models described in Section 3.6.

Theorem 3.12. *Let M be an MO set of linear equations*

$$H\,x + L\,z + c = 0 \tag{3.119}$$

where rank $([H\,L\,c])$ = rank $([H\,L])$ *and let model deviation be described by a vector ϵ such that*

$$H\,x + L\,z + c = \epsilon \tag{3.120}$$

Any residual derived from M can be written

$$r = N_H(Lz + c) \tag{3.121}$$

and the residual response of the model deviation is given by

$$r = N_H\epsilon \tag{3.122}$$

where $N_H[:,\{i\}] \neq 0$ for all $i \in M$.

Proof. Since (3.119) is an MO sets, it has redundancy one. This means that N_H is a vector determined up to a non-zero constant. By multiplying (3.120) with N_H from the left, we get both the residual computational form (3.121) and the fault response (3.122). Since M is an MO set Corollary 3.1 states that all columns in N_H are non-zero, i.e. $N_H[:,\{i\}] \neq 0$ for all $i \in M$. □

According to (3.122) almost any model deviation $\epsilon \neq 0$, i.e. for all $\epsilon \in \mathbb{R}^{|M|}$ except for a line, will be detected with the residual r. To see how Theorem 3.12 is a generalization of Theorem 3.11 apply the result of Theorem 3.12 to the model (3.102) which expressed on matrix form is given in (3.104). By adding the model deviation ϵ, we get

$$\tilde{H}\tilde{x} + \tilde{L}z = \epsilon \tag{3.123}$$

From the n first equations, it follows that $f_i = \epsilon_i$ for $1 \leq i \leq n$. The remaining equations are true in any behavioral mode, i.e. $\epsilon_i = 0$ for all $i \geq n + 1$. By elimination of ϵ in (3.122) we get an expression equivalent to (3.113).

Example 3.20

To give an example of the result of Theorem 3.12 consider the electrical circuit model in (3.3). The faults in this model are not described with fault variables f and it is therefore not possible to use Theorem 3.11. As suggested in Theorem 3.12 a model deviation vector ϵ is added to the model. Since the first equation in (2.44) is true for all behavioral modes, it follows that $\epsilon_1 = 0$. With the matrices H and L defined as in (3.3) the fault influence for the MO sets M_i in (3.88) are given by

$$r_i = N_{H[M_i]}Lz = N_{H[M_i]}\epsilon \tag{3.124}$$

The fault sensitivity is directly given by the coefficients in the matrix $N_{H[M_i]}$. To give an example assume that the resistor 1 is broken and the fault appears such that $V_1 - I_1R_1 = \epsilon_2$ where $\epsilon_2 \neq 0$. Then the residuals r_i will be equal to

$$r_i = N_{H[M_i]}[:, \{2\}]\epsilon_2 \tag{3.125}$$

according to (3.124). This means that the residuals that are based on an MO sets that include equation 2 are the residuals that will be non-zero.

3.10.3 Fault Influence and Null Hypothesis

In the framework used in this thesis, conclusions are drawn only when $r \in \mathcal{R}$ according to (2.7), i.e. if $r \in \mathcal{R}$ then the null hypothesis sys $\in \Phi$ of the test is rejected. To draw the strongest conclusion when $r \in \mathcal{R}$, the set Φ^C should be chosen to be exactly the set of behavioral modes that influence the residual r.

By using the fault influence given in Theorem 3.12 the next theorem shows that the fault influence is given by the model assumptions, i.e. the fault influence is assump$(M)^C$. In the theorem, we will denote the column-span of a matrix A by Im(A).

Theorem 3.13. *Let $\mathbb{M}$ be a diagnosis model with the set $\boldsymbol{M}$ of equations*

$$H\boldsymbol{x} + L\boldsymbol{z} + c = 0 \tag{3.126}$$

where

$$\text{rank}\,([H[\hat{M}]\,L[\hat{M}]\,c[\hat{M}]]) = \text{rank}\,([H[\hat{M}]\,L[\hat{M}]]) \tag{3.127}$$

for all maximal feasible models $\hat{M} \subseteq \boldsymbol{M}$. Let the model deviation be modeled as

$$H\boldsymbol{x} + L\boldsymbol{z} + c = F\epsilon \tag{3.128}$$

where F is a $|\boldsymbol{M}| \times |\boldsymbol{M}|$ matrix defined by

$$F_{ij} = \begin{cases} 1 & \text{if } i = j \text{ and } \text{assump}(e_i) \neq \mathbf{B} \\ 0 & \text{otherwise} \end{cases} \tag{3.129}$$

If

$$\text{Im}(F) \subseteq \text{Im}([H\,L]) \tag{3.130}$$

then for any MO set $M \subseteq \boldsymbol{M}$ of equations and for any residual

$$r = N_{H[M]}L\boldsymbol{z} \tag{3.131}$$

not identically zero and based on M, r is influenced by all behavioral modes in $(\text{assump}(M))^C$ *and no others.*

From Theorem 3.12 we do know that a residual r based an MO set M will be sensitive to all model deviation variables ϵ_i included in the equations M. Condition (3.130) guarantees that there exists a non-zero ϵ such that $r \neq 0$, i.e. the model deviation variables ϵ_i can be seen as independent input signals.

Proof. An MO set M and a residual r are given. We start to prove that r is not influenced by any behavioral mode $b \in \text{assump}(M)$. By the definition of the operator assump, it follows that

$$\text{sys} \in \text{assump}(M) \rightarrow \boldsymbol{z} \in \mathcal{O}(M) \tag{3.132}$$

The observation set $\mathcal{O}(M)$ can according to (3.10) and (3.131) be expressed as

$$\mathcal{O}(M) = \{z|N_{H[M]}Lz = 0\} = \{z|r(z) = 0\} \tag{3.133}$$

The formulas (3.132) and (3.133) imply that

$$\text{sys} \in \text{assump}(M) \rightarrow r = 0 \tag{3.134}$$

i.e. r is not influenced by any behavioral mode $b \in \text{assump}(M)$.

Now, we continue to prove that r is influenced by all behavioral modes $b \notin \text{assump}(M)$. The idea is to take an arbitrary behavioral mode $b \in (\text{assump}(M))^C$ and show that the equations M_b and $r \neq 0$ are consistent. From the definition of the operator assump, it follows for any set $\bar{M}$ that

$$\bar{M} \subseteq M_b \rightarrow b \in \text{assump}(\bar{M})$$

Then, since $b \notin \text{assump}(M)$, it follows that $M \not\subseteq M_b$ or equivalently that $M \setminus M_b \neq \emptyset$. Let $M \setminus M_b$ be denoted by M'. By construction of M', it follows that $b \notin \text{assump}(e)$ for any $e \in M'$. This implies that $\text{assump}(e) \neq \mathbf{B}$ for all $e \in M'$, i.e., $F[M', M']$ is the identity matrix. From (3.128), (3.131), and (3.122), we get the fault influence

$$r = N_{H[M]}F\epsilon \tag{3.135}$$

The only equations $e \in M$ that might be inconsistent, i.e. $\epsilon[\{e\}] \neq 0$, are the equations $e \in M'$. Hence (3.135) can be rewritten as

$$r = N_{H[M]}[:, M']F[M', M']\epsilon[M'] = N_{H[M]}[:, M']\epsilon[M'] \tag{3.136}$$

From (3.130) it follows that

$$\text{Im}(F[M \cup M_b]) \subseteq \text{Im}([H[M \cup M_b]\, L[M \cup M_b]]) \tag{3.137}$$

i.e. for any $\epsilon[M \cup M_b]$ there exist some $\boldsymbol{x}$ and $\boldsymbol{z}$ such that the subset $M \cup M_b$ of the equations in (3.128) is satisfied. Since $M' \cap M_b = \emptyset$, we can choose $\epsilon[M']$ such that $r \neq 0$ in (3.136) and $\epsilon[M_b] = 0$. This implies that we have found an ϵ such that M_b is consistent in (3.128) according to (3.126), and $r \neq 0$. Since b was an arbitrarily chosen behavioral modes such that $b \notin \text{assump}(M)$, it follows that r is influenced by all behavioral modes in $(\text{assump}(M))^C$ and the theorem follows. □

Next two remarks of this theorem will be discussed. First, note that a sufficient condition for (3.130) is that $[H\, L]$ has full row-rank. This means that Theorem 3.13 is not restrictive for diagnosis models where all equations are valid in the no-fault mode. However Theorem 3.13 becomes restrictive in the case with several maximal feasible models.

Second, the fault influence on a residual derived from an MO set M is $(\text{assump}(M))^C$ for the class of models consider in Theorem 3.13. This means that the set Φ in (2.7) should according to the fault influence be chosen as $\Phi = \text{assump}(M)$. This is in agreement with the guideline (2.15) presented in Section 2.2.1.

The next corollary restates Theorem 3.13 for an important special case of the class of models handled in Theorem 3.13, i.e. the type (3.101).

Corollary 3.4. *Let* $\mathbb{M}$ *be a diagnosis model with the set* $\boldsymbol{M}$ *of equations*

$$\begin{array}{ll} c_1 = NF & f_1 = 0 \\ \vdots & \vdots \\ c_n = NF & f_n = 0 \\ & Hx + Lz + Ff = 0 \end{array} \tag{3.138}$$

If

$$F \subseteq \text{Im}([H\,L]) \tag{3.139}$$

then for any MO set $M \subseteq \boldsymbol{M}$ *of equations and for any residual* r *not identically zero and based on* M*,* r *is influenced by all faults in* $(\text{assump}(M))^C$ *and no others.*

Proof. The proof is immediate from Theorem 3.13. □

Now, we will illustrate the result of Theorem 3.13 by three examples. The first and second example show when the condition (3.130) is met and the conclusion of theorem follows. The third example illustrates the case when (3.130) is not fulfilled and it shows why the fault influence $(\text{assump}(M))^C$ is not obtained in this case.

Example 3.21

The continuation of Example 3.19. It can be verified that (3.130) is true either by using the matrices $\bar{H}'$ and $\bar{L}'$ in (3.117) and

$$\bar{F}' = \begin{bmatrix} I_{3\times3} & 0_{3\times2} \\ 0_{2\times3} & 0_{2\times2} \end{bmatrix}$$

or with the matrices H, L, and F in (3.116), i.e.

$$F = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \qquad H = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The assumption of the MO set $\{1, 2, 4, 5\}$ is $\text{assump}(\{1, 2, 4, 5\}) = \phi(c_1 = NF \wedge c_2 = NF)$, i.e. the residual (3.118) is according to Theorem 3.13 or Corollary 3.4 influenced by all faults in

$$(\text{assump}(\{1, 2, 4, 5\}))^C = (\phi(c_1 = NF \wedge c_2 = NF))^C = \phi(c_1 = F \vee c_2 = F)$$

This can also be directly seen in (3.118) and it means that it is possible to detect all faults modes in the complement set of $\text{assump}(\{1, 2, 4, 5\})$.

Example 3.22

For the electrical circuit example with the model (3.95) it holds that $[H\,L]$ has full row rank. This is a sufficient condition for (3.130) and it follows that for any residual r based on an MO set M, $(\text{assump}(M))^C$ are the behavioral modes that r is influenced by.

Next an example shows what can happen when (3.139) is not fulfilled.

Example 3.23
Consider the model

$$\begin{array}{ll} c_1 = NF & f_1 = 0 \\ c_2 = NF & f_2 = 0 \end{array}$$

$$\overbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}^{H} x + \overbrace{\begin{bmatrix} -1 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}}^{L} z + \overbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}}^{F} f = 0 \tag{3.140}$$

The set $\{1, 2, 3, 5\}$ is an MO set and a residual is

$$r = y_1 - y_2 = f_1 - f_2 \tag{3.141}$$

From this expression we would expect that r is influenced by $c_1 = NF \wedge c_2 = F$ which is contained in $(\text{assump}(M))^C$. We will show that this is not true and (3.139) is not fulfilled for this example.

It can be verified that $F \not\subseteq \text{Im}([H\,L])$. It is the column corresponding to f_2 that does not satisfy (3.139) and f_2 cannot be chosen arbitrarily. In fact, the last three equations in (3.140) implies that $f_2 = 0$ in any behavioral mode. By substitution of f_2 in (3.141) the fault influence is given by

$$r = y_1 - y_2 = f_1$$

Hence behavioral modes that influences r are $\phi(c_1 = F) \subset (\text{assump}(M))^C$.

Note that even if the fault influence on a residual based on M is equal to $B \subset (\text{assump}(M))^C$, it is possible to use $\Phi = \text{assump}(M)$ as null hypotheses. The effect is that when the $r \neq 0$ then it is concluded that sys $\in (\text{assump}(M))^C$ but according to the model $r \neq 0$ means that sys $\in B$. However if there is one residual r_i for each model M_i in a set ω that satisfies (2.34) then exactly the same candidates will be obtained if $\Phi_i = \text{assump}(M_i)$ or if Φ_i is equal to the true fault influence of r_i. Hence when designing a sound diagnosis system the exact fault influences on each residual r_i is not important.

3.11 Conclusions

In Chapter 2 we showed that one strategy to construct a diagnosis system was to start with a diagnosis model $\mathbb{M}$ and choose a set $\omega = \{M_1, \ldots, M_n\}$ of rejectable models to test. There, it was also shown that a diagnosis system based on ω can be sound and complete if and only if the set ω fulfills (3.1) for all behavioral modes $b \in \mathbf{B}$.

This chapter has presented theory and algorithms for finding a minimum cardinality solution ω of (3.1) given a diagnosis model $\mathbb{M}$ with linear static equations. A key result is that if ω is chosen to be the set of all feasible MO sets in the diagnosis model $\mathbb{M}$, then ω fulfills (3.1) for all behavioral modes $b \in \mathbf{B}$ according to Theorem 3.10. It has also been shown that it is not in general necessary to include all MO sets in ω to satisfy (3.1) for all behavioral modes $b \in \mathbf{B}$. Theory for selecting MO sets has been developed and a key result for this is the rank condition given in Theorem 3.6. The rank condition is then used in Algorithm 2 for selecting MO sets. The output of the algorithm contains all minimal subsets ω that fulfill (3.1) for all behavioral modes $b \in \mathbf{B}$. A minimal cardinality set of MO sets is then picked out from the set of all minimal sets. Note that this is important because, a minimum cardinality set ω that satisfies (3.1) for all behavioral modes $b \in \mathbf{B}$, corresponds to a sound and complete diagnosis system with the minimum number tests. Several examples of the minimization of the number of tests in the design of a sound and complete diagnosis systems have been given in Section 3.9.

Finally, Theorem 3.13 showed that under a mild rank condition on the diagnosis model and given an MO set M, the behavioral modes that influence any residual derived from M are given by the equation assumptions according to $(\text{assump}(M))^C$. Hence if the rank condition of the model can be verified, then any further fault influence analysis of each residual is not needed.

4

A E A F MO S

In the previous chapter we have shown in Theorem 3.10 that a sound diagnosis system can be based on all feasible MO sets in a diagnosis model $\mathbb{M}$, i.e., all MO sets M in $\mathbb{M}$ such that $\mathrm{assump}(M) \neq \emptyset$. In this chapter an algorithm for computing all MO sets in a linear static or dynamic model is proposed. How to apply the algorithm to dynamic models will be discussed later in Chapter 6. The algorithm developed here will also be the basis for constructing a structural algorithm that can be applied to models containing non-linear differential equations in Chapter 9.

All MO sets in a diagnosis model are in general not feasible MO sets. The algorithm presented in this chapter finds all MO sets in a model, both feasible and non-feasible ones. It is not computationally efficient to find all MO sets and then remove the non-feasible ones to get only the feasible MO sets. In Chapter 5 an algorithm is constructed for finding only feasible MO sets. The algorithm presented here will be the key component in the construction of the algorithm for finding only feasible MO sets.

Section 4.1 and 4.2 introduce key concepts that will be used in the construction of the algorithm. In Section 4.3 a basic algorithm for finding all MO sets will be presented. This algorithm illustrates the basic ideas and then in Section 4.4 further improvements are described. The computational complexity of the algorithm is discussed in Section 4.5. Section 4.6 describes step by step the progress of the algorithm when applied to an illustrative example. Finally the conclusions are drawn in Section 4.7.

4.1 Introduction

We will, as said before, present an algorithm for computing all MO sets in a linear model

$$H\mathbf{x} + L\mathbf{z} = 0 \tag{4.1}$$

The matrices H and L are scalar matrices in the static case and polynomial matrices in the dynamic case. In the continuation of this chapter the examples will be static. Dynamic examples will later be presented in Chapter 6.

The main idea of the algorithm can be explained by looking at the Hasse diagram in Figure 2.2. The algorithm is based on a top-down approach in the sense that we start with the entire model and then reduces the size and the redundancy of the model step by step until an MO set remains. For the Hasse diagram in Figure 2.2 we start with $\{1, 2, 3, 4, 5, 6, 7\}$ and, by for example following the rightmost branch in the graph, $\{3, 4, 5, 7\}$ and then the MO set $\{4, 5\}$ is obtained. How to obtain PO sets with decreasing redundancy is described in the next section.

4.2 The Proper Overdetermined Part

As said in the introduction, the idea is to find MO sets in a PO set by computing a sequence of PO subsets with decreasing redundancy until an MO set is found.

Example 4.1
For the electrical circuit example with PO sets shown in Figure 2.2, the MO set $\{4, 5\}$ can be obtained by starting with the PO set $\{1, 2, 3, 4, 5, 6, 7\}$ with redundancy 3 and then compute the PO subset $\{3, 4, 5, 7\}$ with redundancy 2 and finally compute the MO set $\{4, 5\}$.

This motivates studying how to compute a PO sets $E \subset M'$ with redundancy $\varphi - 1$, given a PO set M' with redundancy $\varphi \geq 2$. Such computation can be divided into the following two steps:

a) Remove an arbitrary equation $e \in M'$ from M' and let $M = M' \setminus \{e\}$.

b) Find a PO set $E \subseteq M \subset M'$ with redundancy $\varphi - 1$.

The rest of this section is organized as follows. First, the existence and uniqueness of the PO set E, specified in step (b), are shown. Then, a method to compute the PO set E given a model M is proposed.

4.2.1 Existence and Uniqueness

The existence and uniqueness of the PO set E, specified in step (b), are shown in the following theorem.

Theorem 4.1. *Given a PO set M' with redundancy $\varphi(M') \geq 2$ and an arbitrary equation $e \in M'$, there exists a unique PO set $E \subseteq M' \setminus \{e\}$ such that $\varphi(E) = \varphi(M') - 1$.*

The uniqueness part of the proof of this theorem will be based on the two following lemmas. In these lemmas we will use the notation of supermodularity, i.e., the redundancy function φ is a super-modular function on $\mathcal{P}(M)$ if

$$\varphi(M_1 \cup M_2) + \varphi(M_1 \cap M_2) \geq \varphi(M_1) + \varphi(M_2) \tag{4.2}$$

for any sets $M_1 \subseteq M$ and $M_2 \subseteq M$.

Lemma 4.1. *If M is a set of equations defined by (3.5) where $[H\, L]$ has full row-rank, then the redundancy function φ is a super-modular function on $\mathcal{P}(M)$.*

Proof. A function is super-modular if (4.2) holds for all subsets $M_1 \subseteq M$ and $M_2 \subseteq M$. Since $[H\,L]$ has full row-rank, it follows that

$$\varphi\,(M') = |M'| - \text{rank}\,(H[M']) \tag{4.3}$$

for all $M' \subseteq M$. By using this, (4.2) is true if

$$\begin{aligned}|M_1 \cup M_2| - \text{rank}\,(H[M_1 \cup M_2]) + |M_1 \cap M_2| - \text{rank}\,(H[M_1 \cap M_2])\\ \geq |M_1| - \text{rank}\,(H[M_1]) + |M_2| - \text{rank}\,(H[M_2])\end{aligned}$$

is true. Simplification of this inequality by using that

$$|M_1 \cup M_2| + |M_1 \cap M_2| = |M_1| + |M_2|$$

gives

$$\text{rank}\,(H[M_1 \cup M_2]) + \text{rank}\,(H[M_1 \cap M_2]) \leq \text{rank}\,(H[M_1]) + \text{rank}\,(H[M_2])$$

which is the definition of the rank-function being sub-modular. The sub-modularity of the rank-function follows from Lemma 3.1 and this completes the proof. □

Lemma 4.2. *Let M be a set of equation such that $\varphi\,(M) \geq 1$ and such that the redundancy function φ is a supermodular function on $\mathcal{P}(M)$. Then there exists a unique PO set $E \subseteq M$ with maximal redundancy among all subsets of M, i.e., $\varphi\,(E) = \varphi\,(M)$.*

A more general formulation of this lemma is stated and proved in (Ore, 1956).

Proof. Theorem 3.3 states that $\varphi\,(E) \leq \varphi\,(M)$ for all subsets $E \subseteq M$. From this and that $M_1 \cup M_2 \subseteq M$ and $M_1 \cap M_2 \subseteq M$, it follows that

$$\varphi\,(M_1 \cup M_2) + \varphi\,(M_1 \cap M_2) \leq 2\,\varphi\,(M) \tag{4.4}$$

Assume that M_1 and M_2 are two PO sets with redundancy $\varphi\,(M) \geq 1$. Since φ is a supermodular function on $\mathcal{P}(M)$, it follows that (4.2) holds for any sets $M_1 \subseteq M$ and $M_2 \subseteq M$. This, (4.2), and (4.4) imply equality in (4.2). From this and the fact that M_1 and M_2 have maximal redundancy among the subsets of M, it follows that

$$\varphi\,(M_1 \cap M_2) = \varphi\,(M_1) = \varphi\,(M_2)$$

From this, $M_1 \cap M_2 \subseteq M_1$, $M_1 \cap M_2 \subseteq M_2$, and M_1 and M_2 are PO sets, i.e.,

$$\varphi\,(E) < \varphi\,(M_i) \tag{4.5}$$

for all proper subsets $E \subset M_i$, we get

$$M_1 = M_1 \cap M_2 = M_2$$

i.e., $M_1 = M_2$. Since M_1 and M_2 was chosen to be two arbitrary PO sets in M with maximal redundancy, it follows that there exists a unique PO set in M with maximal redundancy. □

Now, we are ready to prove Theorem 4.1.

Proof. We start to prove the existence of a PO set $E \subseteq M' \setminus \{e\}$ with redundancy $\varphi(M') - 1$. Since M' is a PO set with $\varphi(M') \geq 2$, it follows that the resulting set $M = M' \setminus \{e\}$ has redundancy $\varphi(M) = \varphi(M') - 1 \geq 1$ according to Theorem 3.4. The existence of a minimal set $E \subseteq M$ with redundancy $\varphi(M)$ is trivial. Then since E is a minimal set with redundancy $\varphi(M) \geq 1$, Theorem 3.3 implies that E is a PO set.

Next the uniqueness is shown. Since M' is a PO set, it follows that the corresponding matrix $[H\,L]$ has full row-rank according to Theorem 3.5. This implies that the matrix $[H[M]\,L[M]]$ corresponding to the set M also has full row rank, i.e., M fulfills the rank condition in Lemma 4.1. This means that the redundancy function φ is a supermodular function on $\mathcal{P}(M)$. From Lemma 4.2 we then get that there exists a unique PO set $E \subseteq M$ with redundancy $\varphi(E) = \varphi(M)$. □

The PO set E specified in step (b), is according to Theorem 4.1 the unique PO set $E \subseteq M$ with maximal redundancy. If we relax the requirement on M such that M is any set of equations such that $[H\,L]$ has full row rank, the supermodularity implies that among the subsets $E \subseteq M$ there exists a unique PO set with maximal redundancy $\varphi(M)$. This PO set will be denoted by M^* and will be called the *proper overdetermined (PO) part of* M. The set M is in (Ore, 1956) suggested to be partitioned into $M^* \cup (M \setminus M^*)$ such that

$$\varphi(M) = \varphi(M^*) \tag{4.6}$$

and

$$\varphi(M \setminus M^*) = 0 \tag{4.7}$$

This equations follows from (4.2) when $M_1 = M \setminus M^*$ and $M_2 = M^*$ and the equation (4.6). This means that M^* contains all redundancy of M and $\mathcal{O}(M^*) = \mathcal{O}(M)$ according to Theorem 3.6.

Looking back on the two computation steps formulated in the beginning of this section, step (b) can be reformulated using the PO part as follows:

b) Find the PO part M^* of M.

Next a method to compute the PO part M^* of M will be developed.

4.2.2 Computing the PO Part

The PO part M^* of M can for example be computed as the set of non-zero columns in an arbitrary basis for the left null-space of H as the next theorem shows.

Theorem 4.2. *If M is a set of equations in the form* (3.5) *where* $[H\,L]$ *has full row-rank, then*

$$M^* = \{e \in M | N_H[:, \{e\}] \neq 0\} \tag{4.8}$$

Before we prove the theorem, let us again consider Example 4.1.

Example 4.2

The PO set $\{3, 4, 5, 7\}$ can be computed from $\{1, 2, 3, 4, 5, 6, 7\}$ as follows. Since

$\{1,2,3,4,5,6,7\}$ is a PO set, it follows from Theorem 3.5 that the corresponding matrix $[H\,L]$ has full row-rank. Hence the redundancy function is supermodular on the family of subsets of $\{1,2,3,4,5,6,7\}$. By removing an equation, lets say the first one, we obtain

$$\varphi\left(\{2,3,4,5,6,7\}\right) = 2 \tag{4.9}$$

From the uniqueness of a PO set with redundancy 2 and that $\varphi\left(\{3,4,5,7\}\right) = 2$, it follows that

$$\{2,3,4,5,6,7\}^* = \{3,4,5,7\} \tag{4.10}$$

From Theorem 4.2 the computation of $\{2,3,4,5,6,7\}^*$ can be done by finding the non-zero columns of

$$N_{H[\{2,3,4,5,6,7\}]} = \begin{bmatrix} 0 & 0 & 1 & 0 & -1 & 0 & R_2 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \end{bmatrix} \tag{4.11}$$

Note that the matrix is zero-padded in accordance with the notation $N_{H[\{2,3,4,5,6,7\}]}$. In this way, it is possible to compute PO sets with decreasing redundancy.

Next, we prove Theorem 4.2.

Proof. Let M' be the set defined by the right-hand side of (4.8). First, we show that $\varphi(M') = \varphi(M)$. From the definition of null-space, it follows that the rows $M \setminus M'$ in H corresponding to zero columns in N_H are linearly independent, i.e.,

$$\text{rank}\,(H) = \text{rank}\,(H[M']) + |M \setminus M'| \tag{4.12}$$

This and that $[H\,L]$ has full row-rank give

$$\begin{aligned} \varphi(M') &= (\text{rank}\,([H[M']\,L[M']]) + |M \setminus M'|) \\ &\quad -(\text{rank}\,(H[M']) + |M \setminus M'|) \\ &= \text{rank}\,([H\,L]) - \text{rank}\,(H) = \varphi(M) \end{aligned}$$

Next, we prove that $\varphi(E) < \varphi(M')$ for all $E \subset M'$. Since all columns in N_H corresponding to rows in M' are non-zero, it means that any row in $H[M']$ is linearly dependent with the other rows, i.e.,

$$\text{rank}\,(H[M']) < \text{rank}\,(H[E]) + |M' \setminus E| \tag{4.13}$$

From this and the fact that $[H\,L]$ has full row-rank, it follows that

$$\begin{aligned} \varphi(E) &= (\text{rank}\,([H[E]\,L[E]]) + |M' \setminus E|) \\ &\quad -(\text{rank}\,(H[E]) + |M' \setminus E|) \\ &< \text{rank}\,([H[M']\,L[M']]) - \text{rank}\,(H[M']) = \varphi(M') \end{aligned}$$

This implies that M' is a PO set and since $\varphi(M') = \varphi(M)$ it also follows that M' has maximum redundancy. Since $[H\,L]$ has full row-rank, there exist a unique PO set with maximum redundancy and this is defined to be the PO part M^* of M. Hence, it follows that $M' = M^*$, i.e., (4.8) is true. □

4.3 Algorithm

In this section we will present an algorithm for finding all MO sets. To illustrate the ideas, a basic version is presented here and then in the next section, improvements are discussed.

4.3.1 Input to the Algorithm

In the algorithm that will be presented next, we start with a set M that will be assume to be a PO set. A typical input is a behavioral model M_b and an example of a behavioral model that is a PO set is the no-fault behavioral model in (3.3) for electrical circuit.

Next we discuss how to obtain a legal input, that is a PO set M, if M_b is not a PO set. If M_b is not a PO set but the corresponding matrix $[H_b\, L_b]$ has full row-rank, then it is easy to obtain M_b^* and use this as input.

In the case when $[H_b\, L_b]$ has not full row-rank, the situation gets more complicated. First, check that the rank deficiency is not caused by a modeling error. If not, there exist two options.

The first option is to compute one PO set $M' \subset M_b$ as follows. A set of linearly independent rows $M \subset M_b$ in $[H_b\, L_b]$ are identified and then the PO part is computed as $M' = M^*$. This set contains all redundancy of M_b and can therefore be used to check the consistency of M_b.

The second option is to compute all PO sets $M \subseteq M_b$ with maximal redundancy and then apply the following algorithm to each of these PO sets. A disadvantage with this is that the number of MO sets will be greater compared to the number of MO sets obtained in the first way.

4.3.2 Basic Algorithm

The algorithm will be based on Lemma 3.2, Theorem 3.4, and the following lemma.

Lemma 4.3. *If M is a set of equations in the form* (3.5) *where* $[H\, L]$ *has full row-rank, $E \subseteq M$ is a PO set, and $e \in M \setminus E$, then*

$$E \subseteq (M \setminus \{e\})^* \tag{4.14}$$

Proof. The left null space of E is according to $E \subseteq M \setminus \{e\}$ a subspace of the left null space of $M \setminus \{e\}$. From the subspace relation, it follows that

$$N_{H[E]} = \gamma N_{H[M \setminus \{e\}]} \tag{4.15}$$

where γ is a full row-rank matrix. Since $[H\, L]$ has full row-rank, it follows from Theorem 4.2 that $(M \setminus \{e\})^*$ are the non-zero columns in an arbitrary basis for the left null-space of $H[M \setminus \{e\}]$. This implies that

$$N_{H[E]}[:, M \setminus (M \setminus \{e\})^*] = \gamma N_{H[M \setminus \{e\}]}[:, M \setminus (M \setminus \{e\})^*] = 0 \tag{4.16}$$

Since E is a PO set, Corollary 3.1 implies that $N_{H[E]}[:, \{e\}] \neq 0$ for all $e \in E$. This and (4.16), imply that $E \cap (M \setminus (M \setminus \{e\})^*) = \emptyset$ which is equivalent to (4.14). □

Theorem 3.4 reveals how the redundancy decreases when one equation is removed. It follows from this lemma that if we start with any PO set of equations we can alternately remove equations and computing the overdetermined part until the redundancy becomes one. We have then found a MO set, according to Lemma 3.2. Finally, Lemma 4.3 implies that an arbitrary PO set, and therefore also any MO set, can be obtained recursively this way. By using this principle in combination with a complete search the algorithm becomes as follows. The input set M is assumed to be a PO set.

Algorithm 3. `FindMO`(M)

```
if $\varphi(M) = 1$ then
    $\mathcal{M}_{MO} := \{M\}$;
else
    $\mathcal{M}_{MO} := \emptyset$;
    for each equation $e$ in $M$ do
        $M' := (M \setminus \{e\})^*$;
        $\mathcal{M}_{MO} := \mathcal{M}_{MO} \cup$ FindMO($M'$);
    end for
end if
return $\mathcal{M}_{MO}$
```

From the discussion above, it follows that the sets found in $\mathcal{M}_{MO}$ are MO sets and that all MO sets are found.

Example 4.3

To illustrate the steps in the algorithm, consider a PO set $M = \{e_1, e_2, e_3, e_4\}$ with the following H-matrix:

$$H = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 2 \end{bmatrix} \tag{4.17}$$

Remember that the L-matrix is irrelevant when computing the redundancy of PO sets, because the matrix $[H\,L]$ has full row-rank. The redundancy of the set M in (4.17) is 2. When entering the algorithm, e_1 is removed and the set M' becomes $(M\backslash\{e_1\})^* = \{e_3, e_4\}$. In this case $\varphi(M') = 1$ and the equation set is saved as an MO set in $\mathcal{M}_{MO}$. Then e_2 is removed and $M' = (M\backslash\{2\})^* = \{3, 4\}$. This means that the same MO set is found once again. Next e_3 is removed and the MO set $\{e_1, e_2, e_4\}$ is found. Finally e_4 is removed and the MO set $\{e_1, e_2, e_3\}$ is found.

Since the same MO set $\{e_3, e_4\}$ is found twice in the example, we can suspect that the algorithm is not optimal in terms of efficiency. The next section will therefore present improvements in order to increase the efficiency.

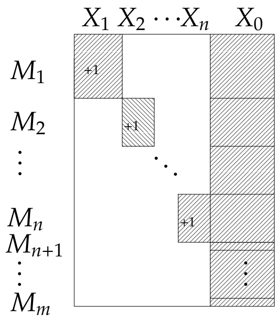

Figure 4.1: A decomposition of the H-matrix of a PO set.

4.4 Improvements

A straightforward improvement is of course to prohibit that any of the MO sets are found more than once. Another and more sophisticated improvement is that sets of equations can be lumped together in order to reduce the size and the complexity of the H-matrix. This reduction preserves redundancy and it is therefore possible to use the reduced matrix to find all MO sets in the original matrix. However, the reduction can also introduce numerical problems and it will therefore be considered as optional.

4.4.1 A Canonical Partition of PO Sets

The reduction is based on a unique partition of a PO set. The partition is based on the matrix H corresponding to a set of equations M and an illustration of the partition is shown in Figure 4.1. The partition can be defined as follows. Let R be a relation on the set M of equations defined by $(e, e') \in R$ if

$$e \notin (M \setminus \{e'\})^* \tag{4.18}$$

Lemma 4.4. *The relation R is an equivalence relation on a PO set M.*

Proof. Now we show that R is an equivalence relation on M, i.e., that R is reflexive, symmetric, and transitive. It follows directly from the definition that R is reflexive. If $(e, e') \in R$, then it follows from (4.18) that $(M \setminus \{e'\})^* \subseteq M \setminus \{e\}$. Lemma 4.3, with E replaced by $(M \setminus \{e'\})^*$, implies that

$$(M \setminus \{e'\})^* \subseteq (M \setminus \{e\})^* \tag{4.19}$$

Since M is a PO set, Theorem 3.4 implies that $\varphi(M \setminus \{e'\}) = \varphi(M \setminus \{e\})$. From this and (4.6), we get that

$$\varphi((M \setminus \{e'\})^*) = \varphi((M \setminus \{e\})^*) \tag{4.20}$$

Assume that we have inequality in (4.19). Then, let $\hat{e} \in (M \setminus \{e\})^* \setminus (M \setminus \{e'\})^*$. The set $((M \setminus \{e\})^* \setminus \{\hat{e}\})^*$ is according to Lemma 4.3 a superset to $(M \setminus \{e'\})^*$, but have lower redundancy than $(M \setminus \{e'\})^*$ according to (4.20) and Theorem 3.4. This is a contradiction according to Theorem 3.3 and it follows that $(M \setminus \{e'\})^* = (M \setminus \{e\})^*$.

Hence $(e, e') \in R$ and R is therefore symmetric. Furthermore if $(e_1, e_2) \in R$ and $(e_2, e_3) \in R$, then it holds that $(M \setminus \{e_1\})^* = (M \setminus \{e_2\})^* = (M \setminus \{e_3\})^*$, which implies that R is transitive. The relation R is therefore an equivalence relation on M. □

For a linear space $\mathcal{A}$, the orthogonal complement will be denoted by $\mathcal{A}^\perp$. The set M can then be partitioned into disjoint equivalence classes M_i where $i \in \{1, 2, \dots, m\}$. For each equation set M_i, a matrix X_i is defined as a row-basis for the space

$$\mathrm{sp}(H) \cap (\mathrm{sp}(H[M \setminus M_i]))^\perp \tag{4.21}$$

and X_0 is a row-basis for

$$\mathrm{sp}(H) \cap (\mathrm{sp}(\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{bmatrix}))^\perp$$

By this construction, any row in X_i, where $i \neq 0$, is linearly independent with the rows $M \setminus M_i$ in H. No filling in the intersection between M_i and X_j for $i \neq j \neq 0$ is used to indicate this in the figure. Furthermore M_i and X_i is related as follows.

Corollary 4.1. *If M is a PO set, then for all its equivalence classes M_i defined by (4.18), it holds that*

$$|M_i| = \mathrm{rank}\,(X_i) + 1 \tag{4.22}$$

for all $1 \le i \le m$.

Corollary 4.1 states that there is one more equation in M_i than the dimension of the space defined by X_i in each block. Furthermore for $n + 1 \le i \le m$ in the figure, M_i has cardinality 1 and X_i is the zero-space, i.e., $X_i = \{0\}$.

Proof. Let M_i be an arbitrary equivalence class which according to the decomposition implies that for any $e \in M_i$, $(M \setminus \{e\})^* = M \setminus M_i$. Since M is a PO set, $[H\,L]$ has full row-rank and we get that

$$\varphi\,(M) - \varphi\,((M \setminus \{e\})^*) = (|M| - \mathrm{rank}\,(H)) - (|M \setminus M_i| - \mathrm{rank}\,(H[M \setminus M_i]))$$

This can be written as

$$\varphi\,(M) - \varphi\,((M \setminus \{e\})^*) = |M_i| - (\dim(\mathrm{sp}(H)) - \dim(\mathrm{sp}(H[M \setminus M_i])))$$

The last terms can be written

$$\dim(\mathrm{sp}(H)) - \dim(\mathrm{sp}(H[M \setminus M_i])) = \dim(\mathrm{sp}(H) \cap (\mathrm{sp}(H[M \setminus M_i]))^\perp) \tag{4.23}$$

and then we obtain

$$\varphi\,(M) - \varphi\,((M \setminus \{e\})^*) = |M_i| - (\dim(\mathrm{sp}(H) \cap (\mathrm{sp}(H[M \setminus M_i]))^\perp)$$

This and (4.21) imply that

$$\varphi\,(M) - \varphi\,((M \setminus \{e\})^*) = |M_i| - \dim(\mathrm{sp}(X_i)) = |M_i| - \mathrm{rank}\,(X_i)$$

Then Theorem 3.4 and (4.6) imply (4.22) and this proves the corollary. □

By using this partition into equivalence classes M_i, all PO sets can be represented as follows.

Theorem 4.3. *If $E \subseteq M$ is a PO set, then E is a union of equivalence classes defined by (4.18), i.e.,*

$$E = \bigcup_{i \in I} M_i$$

where $I \subseteq \{1, 2, \ldots, m\}$.

Proof. The theorem holds if and only if $E \subseteq M$ and $E \cap M_i \neq \emptyset$ implies that $M_i \subseteq E$. Assume that $E \subseteq M$, $E \cap M_i \neq \emptyset$, and that there exists an $e \in M_i \setminus E \subseteq M \setminus E$. From Lemma 4.3, it follows that $E \subseteq (M \setminus \{e\})^*$. This and the definition of M_i imply that $E \subseteq M \setminus M_i$, which contradicts the assumption and the lemma follows. □

4.4.2 Analytical Reduction

A new linear static system can be formed with each equivalence class M_i of M corresponding to a new equation. The equation corresponding to equivalence class M_i is

$$N_i \left(H[M_i]x + L[M_i]z\right) = 0 \tag{4.24}$$

where N_i is a row vector that fulfill

$$N_i \, H[M_i] \, X_i^T = 0 \tag{4.25}$$

and

$$N_i \not\equiv 0 \tag{4.26}$$

If (4.25) is used to compute N_i, then X_i must be known. However, it is not necessary to compute X_i to compute N_i. The following lemma describes a better way to compute N_i directly from the matrix H, i.e., without first computing the matrix X_i.

Lemma 4.5. *If j is any row such that $N_H[j, M_i] \not\equiv 0$, then $N_i = N_H[j, M_i]$ fulfills* (4.25) *and* (4.26).

Proof. From

$$N_H H = 0 \tag{4.27}$$

it follows that

$$N_H H \, X_i^T = 0 \tag{4.28}$$

This can be expanded to

$$N_H[:, M \setminus M_i] \, H[M \setminus M_i] \, X_i^T + N_H[:, M_i] \, H[M_i] \, X_i^T = 0 \tag{4.29}$$

From (4.21), it follows that $H[M \setminus M_i] \, X_i^T = 0$, i.e., the first term in (4.29) is zero. Hence the second term in (4.29) must also be zero, i.e.,

$$N_H[:, M_i] \, H[M_i] \, X_i^T = 0 \tag{4.30}$$

Since j is an arbitrary row in $N_H[:, M_i]$ such that $N_H[j, M_i] \not\equiv 0$, then $N_i = N_H[j, M_i]$ fulfills both (4.25) and (4.26). □

An equation of the type (4.24) will be called a lumped equation. The set of all lumped equations becomes

$$\overbrace{\begin{bmatrix} N_1 & 0 & \cdots & 0 \\ 0 & N_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & N_m \end{bmatrix}}^{N}(Hx + Lz) = 0 \tag{4.31}$$

and is called the lumped system. The rows in the lumped system will simply be denoted by $\{M_i | i \in \{1, 2, \ldots, n\}\}$. The matrix N has full row-rank. Since M is assumed to be a PO set, $[H\,L]$ has full row rank and this implies that for the lumped system, the matrix $\begin{bmatrix} NH & NL \end{bmatrix}$ has full row-rank. The redundancy is therefore only dependent of NH.

Example 4.4
For example, the lumping of (4.17) is

equivalence class M_i	unknown x_2
$\{e_1, e_2\}$	$0 \quad 1$
$\{e_3\}$	$0 \quad 1$
$\{e_4\}$	$0 \quad 2$

and $X_0 = [0\,1]$, $X_1 = [1\,0]$, and $X_2 = X_3 = []$. The first row in the lumped matrix NH is obtained by multiplying $H[\{e_1, e_2\}]$ from the left with $N_1 = [-1\,1]$. Note that it is only equivalence classes of cardinality greater than one that give a reduction. An interpretation of this reduction is that the two first equations are used to eliminate the space spanned by X_1, in this case the unknown x_1.

In the lumped matrix NH, each equivalence class of M corresponds to one row, and the definitions of PO set, MO set, and redundancy are thereby extended to lumped matrices. In the example above we have $\varphi\left(\{\{e_1, e_2\}, \{e_3\}, \{e_4\}\}\right) = 2$. The redundancy for the lumped and the original system are always the same.

The reduction is justified by the following theorem, which shows that there is a one-to-one correspondence between the PO sets in the original and in the lumped system and that the reduced matrix NH can be used to find all PO sets in the original matrix H.

Theorem 4.4. *The set $\{M_i\}_{i \in I}$ is a PO set in the lumped system if and only if $\cup_{i \in I} M_i$ is a PO set in the original equation system. Further, it holds that*

$$O(\{M_i\}_{i \in I}) = O(\cup_{i \in I} M_i) \tag{4.32}$$

To prove this theorem the following lemma will be used.

Lemma 4.6. *If M is a PO set and $\{M_i\}_{i \in I}$ its equivalence classes, then*

$$\varphi\left(\cup_{i \in I'} M_i\right) = \varphi\left(\{M_i\}_{i \in I'}\right) \tag{4.33}$$

for all $I' \subseteq I$.

Proof. The redundancy of $\cup_{i \in I'} M_i$ is

$$\begin{aligned} \varphi\left(\cup_{i \in I'} M_i\right) &= \left|\cup_{i \in I'} M_i\right| - \operatorname{rank}\left(H[\cup_{i \in I'} M_i]\right) \\ &= \left|\cup_{i \in I'} M_i\right| - \dim(\operatorname{sp}(H[\cup_{i \in I'} M_i])) \end{aligned}$$

By using the notation of the decomposition described in Section 4.4.1 we know that the rows in X_i for $i = 0, 1, \ldots, m$ forms a row basis for H and we get that

$$\varphi\left(\cup_{i \in I'} M_i\right) = \left|\cup_{i \in I'} M_i\right| - \sum_{i \in I'} \dim(\operatorname{sp}(X_i)) - \dim(\operatorname{sp}(X_0) \cap \operatorname{sp}(H[\cup_{i \in I'} M_i])) \quad (4.34)$$

which can be rewritten as

$$\varphi\left(\cup_{i \in I'} M_i\right) = \sum_{i \in I'} (|M_i| - \operatorname{rank}(X_i)) - \dim(\operatorname{sp}(X_0) \cap \operatorname{sp}(H[\cup_{i \in I'} M_i])) \quad (4.35)$$

Corollary 4.1 states that $|M_i| = \operatorname{rank}(X_i) + 1$ for all $i \in I$, and consequently that

$$\varphi\left(\cup_{i \in I'} M_i\right) = |I'| - \dim(\operatorname{sp}(X_0) \cap \operatorname{sp}(H[\cup_{i \in I'} M_i])) \quad (4.36)$$

which is equal to $\varphi\left(\{M_i\}_{i \in I'}\right)$. □

For a model M' where $[H' \; L']$ has full row-rank we will later also use the following alternative characterization of PO sets, based on the definition of PO part. We know that if $M^* \neq \emptyset$ then M^* is a PO set and contrary, for a PO set M it follows that

$$M = M^* \quad (4.37)$$

according to (4.8) and Corollary 3.1. Hence $M \neq \emptyset$ is a PO set if and only if (4.37) holds.

Next, the proof of Theorem 4.4 follows.

Proof. We start to prove that $\{M_i\}_{i \in I}$ is a PO set if and only if $\cup_{i \in I} M_i$ is a PO set. Assume that $\cup_{i \in J} M_i$ is a PO set. Then it follows from Definition 3.3 that

$$\varphi\left(\cup_{i \in J'} M_i\right) < \varphi\left(\cup_{i \in J} M_i\right) \quad (4.38)$$

for all $J' \subset J$. From Lemma 4.6, it then follows that

$$\varphi\left(\{M_i\}_{i \in J'}\right) < \varphi\left(\{M_i\}_{i \in J}\right) \quad (4.39)$$

for all $J' \subset J$. Hence $\{M_i\}_{i \in J}$ is a minimal set with redundancy $\varphi\left(\{M_i\}_{i \in J}\right)$, i.e., $\{M_i\}_{i \in J}$ is a PO set according to Definition 3.3.

Now, we will show the reverse implication. Assume that $\{M_i\}_{i \in J}$ is a PO set. If $M' \subset \cup_{i \in J} M_i$, then

$$M' \supseteq (M')^* = \cup_{i \in J'} M_i \quad (4.40)$$

for some $J' \subset J$ according to Theorem 4.3. Since $\{M_i\}_{i \in J}$ is a PO set, it follows from Lemma 4.6 that

$$\varphi\left(\cup_{i \in J} M_i\right) = \varphi\left(\{M_i\}_{i \in J}\right) > \varphi\left(\{M_i\}_{i \in J'}\right) = \varphi\left(\cup_{i \in J'} M_i\right) \quad (4.41)$$

From (4.37) and (4.40), it follows that

$$\varphi\left(\cup_{i \in J'} M_i\right) = \varphi\left((M')^*\right) = \varphi\left(M'\right) \quad (4.42)$$

The inequality (4.41) and the equality (4.42) imply that $\cup_{i\in J}M_i$ is a minimal set with redundancy $\varphi\left(\cup_{i\in J}M_i\right)$, i.e., $\cup_{i\in J}M_i$ is a PO set according to Definition 3.3.

Now, we prove (4.32). Let H and L be the matrices corresponding to the equation set $\cup_{i\in I}M_i$ and let

$$N(Hx + Lz) = 0$$

be the lumped equations $\{M_i\}_{i\in I}$. The equality (4.32) holds if and only if

$$\text{sp}(N_H L) = \text{sp}(N_{H_l} N L) \tag{4.43}$$

where $H_l = N\,H$. Since

$$N_{H_l} N H = 0 \tag{4.44}$$

it follows that

$$\text{sp}(N_H L) \supseteq \text{sp}(N_{H_l} N L) \tag{4.45}$$

From Lemma 4.6, we get that

$$\dim(\text{sp}(N_H L)) = \varphi\left(\cup_{i\in I}M_i\right) = \varphi\left(\{M_i\}_{i\in I}\right) = \dim(\text{sp}(N_{H_l} N L)) \tag{4.46}$$

which together with (4.45) imply (4.43) which in turn implies (4.32). □

4.4.3 Improved Algorithm

A drawback with Algorithm 3, presented in Section 4.3, is that some of the MO sets are found more than once. There are two reasons why this happens and these can be illustrated using the following example.

Example 4.5
Consider

$$H = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 2 \end{bmatrix} \tag{4.47}$$

where the rows correspond to the equations $\{e_1, \ldots, e_5\}$. First, the same PO set $\{e_3, e_4, e_5\}$ is obtained if e_1 is removed. Second, the same MO set is obtained if the order of equation removal is permuted. For example, the MO set $\{e_4, e_5\}$ is obtained if first e_1 or e_2 and then e_3 is removed but also if the order of removal is reversed.

To illustrate how these two problems are handled in an improved algorithm to be presented later, we use Example 4.5.

To avoid the first problem, the lumping described in previous section is used. Initially we start with the set $M = \{e_1, e_2, e_3, e_4, e_5\}$ and e_1 and e_2 are lumped together and the resulting set is $\mathcal{S}' = \{\{e_1, e_2\}, \{e_3\}, \{e_4\}, \{e_5\}\}$. Similar to the basic algorithm we remove one equivalence class at a time from $\mathcal{S}'$ and make a recursive call which returns all MO sets in the input set.

To avoid the problem with permuted removal order an additional input set $\mathcal{E}'$ is used which contains the equivalence classes that are allowed to be removed in the recursive calls.

Example 4.6
In Example 4.5, we start initially with the set $\mathcal{E}' = \mathcal{S}'$, meaning that all equivalence classes are allowed to be removed. In the first step the equivalence class

$\{e_1, e_2\}$ is removed and a recursive call is made with the input sets

$$\mathcal{S}' \setminus \{\{e_1, e_2\}\} \text{ and } \{\{e_3\}, \{e_4\}, \{e_5\}\}$$

To prevent that the order of removal is permuted we remove the equivalence class $\{e_1, e_2\}$ permanently from $\mathcal{E}'$. In the following step the equivalence class $\{e_3\}$ is removed and the inputs are

$$\mathcal{S}' \setminus \{\{e_3\}\} \text{ and } \{\{e_4\}, \{e_5\}\}$$

Following the same principles, the final calls are made with the input sets

$$\mathcal{S}' \setminus \{\{e_4\}\} \text{ and } \{\{e_5\}\},$$
$$\mathcal{S}' \setminus \{\{e_5\}\} \text{ and } \emptyset$$

To apply these ideas in all steps in the recursive algorithm, the lumping strategy has to be extended to subsets of previously lumped systems. Equivalence classes are then lumped together into new sets of equations by taking the union of the sets in the equivalence class. We illustrate this with a new example.

Example 4.7
Assume that we start with six equations and that e_2 and e_3 are lumped together and the following H-matrix has been obtained:

$$\begin{array}{c|c}
\text{equivalence classes} & H\text{-matrix} \\ \hline
\begin{array}{c} \{e_1\} \\ \{e_2, e_3\} \\ \{e_4\} \\ \{e_5\} \\ \{e_6\} \end{array} & \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 2 \end{bmatrix}
\end{array} \tag{4.48}$$

In the first recursive call, $\{e_1\}$ is removed and the matrix corresponding to the remaining part has the same matrix as in (4.17). Now, it holds that

$$[\{e_2, e_3\}] = [\{e_4\}] = \{\{e_2, e_3\}, \{e_4\}\}$$

where $[E]$ denotes the equivalence class containing E. The sets $\{e_2, e_3\}$ and $\{e_4\}$ are therefore lumped together into the set $\{e_2, e_3, e_4\}$.

Given a model $\mathcal{S}$ and corresponding set $\mathcal{E}$, the lumped system $\mathcal{S}'$ is constructed as described above, and a problem is then how to form the new set $\mathcal{E}'$ of equivalence classes that are allowed to be removed in the new system $\mathcal{S}'$. The following principle will be used. An equivalence class in $\mathcal{S}'$ is allowed to be removed, i.e., the equivalence class belongs to $\mathcal{E}'$, if and only if it is a union of classes that are all allowed to be removed in $\mathcal{S}$, i.e., it belongs to $\mathcal{E}$. It will be shown that, in this way, all MO sets are found once and only once.

It is sufficient to only lump equivalence classes with an non-empty intersection with $\mathcal{E}$ and this is used in the algorithm. To do this partial lumping we will use the notation

$$\mathcal{S}' = \texttt{Lump}(E, \mathcal{S})$$

in the algorithm to denote that only the equivalence classes $[E]$ in the input $\mathcal{S}$ are lumped forming a new set of equivalence classes $\mathcal{S}'$ and the corresponding lumped system. The improved algorithm take $\mathcal{S} := \{\{e\}|e \in M\}$ where M is PO set and $\mathcal{E} = \mathcal{S}$ as input sets, and can formally be written as follows.

```
**Algorithm 4.** FindMO($\mathcal{S},\mathcal{E}$)
if $\varphi(\mathcal{S}) = 1$ then
    $\mathcal{M}_{MO} := \{\cup_{E\in\mathcal{S}} E\}$;
else
    $\mathcal{E}' := \emptyset; \mathcal{S}' := \mathcal{S}$;
    % Lump the system $\mathcal{S}'$ and create $\mathcal{E}'$
    while $\mathcal{E} \neq \emptyset$ do
        Select an $E \in \mathcal{E}$;
        $\mathcal{S}' :=$ Lump($E,\mathcal{S}'$);
        if $[E] \subseteq \mathcal{E}$ then
            $\mathcal{E}' := \mathcal{E}' \cup \{\cup_{E'\in[E]} E'\}$;
        end if
        $\mathcal{E} := \mathcal{E} \setminus [E]$;
    end while
    $\mathcal{M}_{MO} := \emptyset$;
    % Make the recursive calls
    while $\mathcal{E}' \neq \emptyset$ do
        Select an $E \in \mathcal{E}'$
        $\mathcal{E}' := \mathcal{E}' \setminus \{E\}$;
        $\mathcal{M}_{MO} := \mathcal{M}_{MO} \cup$ FindMO($\mathcal{S}' \setminus \{E\},\mathcal{E}'$);
    end while
end if
return $\mathcal{M}_{MO}$
```

The algorithm is justified by the following result.

Theorem 4.5. *If Algorithm 4 is applied to a PO set M of equations, then each MO set contained in M is found once and only once.*

Proof. First, it is shown that each MO set is found at least once. Let $E \subseteq M$ be an arbitrary MO set. A branch, of the recursive tree, that results in this MO set can be obtained in the following way: In each recursive step, chose the *first* branch where an equivalence class not included in E is removed. It follows from Lemma 4.3 and Theorem 4.4 that by following this branch, a sequence of decreasing PO sets all containing E is obtained. Hence the MO set E is found this way.

Finally, it is shown that the same MO set E can not be found if we deviate from the branch described above, i.e., that the MO set E is found only once. In each recursive step, in all branches that precede this branch, only equivalence classes contained in E have been removed. Therefore, these branches do not result in the set E. On the other hand all succeeding branches contain the *first* equivalence class $\hat{E}$ not contained in E, i.e., the class removed in the branch that gives the set E. This follows from the fact that $\hat{E}$ has been removed from $\mathcal{E}$ and is not allowed to be removed. Furthermore in all lumped systems in these branches, $\mathcal{E}'$ is constructed such that $\hat{E}$ is an equivalence class not contained in $\mathcal{E}'$. Hence, the branch described above is the only branch that results in the MO set E. This completes the proof. □

4.4.4 Algorithm without Lumping

As mentioned in the beginning of this section the lumping can introduce numerical problems. To avoid these, it is possible to omit the lumping of H but still use the idea of equivalence classes. In Algorithm 4, the only two algebraic computations are the initial PO set computation and the lumping. Of these two computations, it is only the lumping that uses previously lumped equations. Therefore the modification corresponds to replace $\texttt{Lump}(E, \mathcal{S}')$ with a function

$$\mathcal{S}' = \texttt{EquivalenceClasses}(E, \mathcal{S}) \tag{4.49}$$

that computes the equivalence classes but does not lump equations. The output $\mathcal{S}'$ is computed as follows. First

$$M_l = (\cup_{E' \in \mathcal{S}} E') \setminus (\cup_{E' \in \mathcal{S} \setminus E} E')^* \tag{4.50}$$

is computed. Then the equivalence classes $E' \in \mathcal{S}$ where $E' \subseteq M_l$ are replaced with the single set M_l. The resulting $\mathcal{S}$ is equal to the output $\mathcal{S}'$. Note that the PO part computation in (4.50) is applied to a subset of the original equations M, i.e., a subset of rows of the original matrix H. Hence all algebraic computations are done using the matrix H.

Example 4.8
As an example consider

$$H = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

where the rows correspond to the equations $M = \{e_1, \ldots, e_6\}$. Let $\mathcal{S}' = \{\{e_1, e_2\}, \{e_3, e_4\}, \{e_5\}, \{e_6\}\}$ and $E = \{e_1, e_2\}$ be the input sets to $\texttt{EquivalenceClasses}$. The computation (4.50) is for this example

$$M_l = M \setminus \{e_3, e_4, e_5, e_6\}^* = M \setminus \{e_5, e_6\} = \{e_1, e_2, e_3, e_4\}$$

Note that the PO set in the example was computed by using the matrix $H[\{e_3, e_4, e_5, e_6\}]$ which is a subset of rows in H. By omitting the lumping, null space computations of matrices obtained by previous null-space computations are avoided and this avoids potential numerical problems.

4.5 Computational Complexity

For a fixed order of redundancy, the computational complexity is polynomial in the number of equations. This follows from the fact that, in the algorithm, the number of recursive calls is equal to the number of PO sets. The worst case is when no equations are lumped. In this case the PO sets are all subsets with cardinality strictly greater than the number of unknowns in the original model. By similar computations as done in Section 3.7, it follows that the number of such sets grows polynomially in the number of equations. Furthermore the computational complexity to obtain the set M^*, has the same computational complexity as making a QR-factorization of the matrix H. The QR-factorization is polynomial in the number of equations. For a fixed number of unknowns, the complexity of the improved algorithms are exponential in the number of equations. However, this situation is, as mention in Section 3.7, not common in application areas such as the automotive industry.

4.6 Example: The Electrical Circuit

In this section we briefly describe some of the steps when Algorithm 4 is applied to the electrical circuit presented in Chapter 3.

Consider the Hasse diagram in Figure 2.2 and let the nodes be enumerated from left to right and then from the top to the bottom. In the first subroutine call we have the inputs:

$$\mathcal{S} = \mathcal{E} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}\}$$

i.e., we enter node 1. Then the lumping of $\mathcal{E}$ results in

$$\mathcal{S}' = \mathcal{E}' = \{\{1, 2, 6\}, \{3\}, \{4\}, \{5\}, \{7\}\}$$

Assume then that $E = \{1, 2, 6\} \in \mathcal{E}'$ is selected. This means that

$$\mathcal{E}' = \{\{3\}, \{4\}, \{5\}, \{7\}\}$$

and the inputs to the next subroutine call become

$$\mathcal{S} = \mathcal{E} = \{\{3\}, \{4\}, \{5\}, \{7\}\}$$

i.e., we enter node 6. Reduction of this is

$$\mathcal{S}' = \mathcal{E}' = \{\{3, 7\}, \{4\}, \{5\}\}$$

Assume next that $E = \{3, 7\} \in \mathcal{E}'$ is selected, This implies

$$\mathcal{E}' = \{\{4\}, \{5\}\}$$

and the inputs to the recursive call become

$$\mathcal{S} = \mathcal{E} = \{\{4\}, \{5\}\}$$

i.e., node 14 is entered. The model $\mathcal{S}$ is identified to be a MO sets. Hence

$$\mathcal{M}_{MO} := \{\cup_{E \in \mathcal{S}} E\} = \{4, 5\}$$

Now, backtracking to node 6 and the situation is

$$\begin{aligned}\mathcal{S}' &= \{\{3,7\},\{4\},\{5\}\}\\ \mathcal{E}' &= \{\{4\},\{5\}\}\end{aligned}$$

Then we select a new equivalence class say $\{4\} \in \mathcal{E}'$ and get

$$\begin{aligned}\mathcal{S}' &= \{\{3,7\},\{4\},\{5\}\}\\ \mathcal{E}' &= \{\{5\}\}\end{aligned}$$

The input arguments to the recursive call are

$$\begin{aligned}\mathcal{S} &= \{\{3,7\},\{5\}\}\\ \mathcal{E} &= \{\{5\}\}\end{aligned}$$

This is identified as an MO set, i.e., $\{3,5,7\}$ is found. Returning to node 6 we have that

$$\begin{aligned}\mathcal{S}' &= \{\{3,7\},\{4\},\{5\}\}\\ \mathcal{E}' &= \{\{5\}\}\end{aligned}$$

and

$$\mathcal{M}_{MO} := \{\{4,5\}\} \cup \{\{3,5,7\}\}$$

In the final recursive call we select $\{5\} \in \mathcal{E}'$ and

$$\begin{aligned}\mathcal{S} &= \{\{3,7\},\{4\}\}\\ \mathcal{E} &= \emptyset\end{aligned}$$

$\{3,4,7\}$ is identified to be an MO set. Returning to node 6, we get

$$\mathcal{M}_{MO} := \{\{4,5\},\{3,5,7\}\} \cup \{\{3,4,7\}\}$$

Since

$$\begin{aligned}\mathcal{S}' &= \{\{3,7\},\{4\},\{5\}\}\\ \mathcal{E}' &= \emptyset\end{aligned}$$

i.e., $\mathcal{E}$ is empty, we backtrack to the node 1 and we have

$$\begin{aligned}\mathcal{S}' &= \{\{1,2,6\},\{3\},\{4\},\{5\},\{7\}\}\\ \mathcal{E}' &= \{\{3\},\{4\},\{5\},\{7\}\}\end{aligned}$$

Continuing in the same way, we select $\{3\} \in \mathcal{E}'$ and enter node 5 with

$$\begin{aligned}\mathcal{S} &= \{\{1,2,6\},\{4\},\{5\},\{7\}\}\\ \mathcal{E} &= \{\{4\},\{5\},\{7\}\}\end{aligned}$$

When the lumping is performed, it turns out that

$$[\{7\}] = \{\{1,2,6\},\{7\}\}$$

i.e.,

$$\mathcal{S}' = \{\{1,2,6,7\},\{4\},\{5\}\}$$

It follows that $[\{7\}] \not\subseteq \mathcal{E}'$ and this implies that

$$\mathcal{E}' = \{\{4\},\{5\}\}$$

In this way the set $\{1,2,6,7\}$ is not removed. If this equivalence class would be removed the resulting set would be $\{\{4\},\{5\}\}$, i.e., we would enter node 14 again. By selecting the equivalence class including the lowest equation first, the nodes will be traversed in the following order: 1, 6, 14, 13, 12, 5 as we already have explained and then 11, 10, 4, 9, 8, 3, 7, and finally 2.

4.7 Conclusions

An algorithm for computing all MO sets of equations has been developed. There are three main ideas that are used in the algorithm. First, it is based on a top-down approach as described in Section 4.3. Second, an analytical reduction is used where subsets of equations can be lumped together in order to reduce the size of the model. Third and last, it is prohibited that any MO set is found more than once. For a fixed order of redundancy, the computational complexity, of the algorithm, is polynomial in the number of equations.

5

A A F F MO S

If a sound diagnosis system for a diagnosis model $\mathbb{M}$ is to be designed based on a set $\omega = \{M_1, \dots, M_n\}$ of models, it has been shown in Chapter 2 that ω must fulfill (2.34) for all behavioral modes in $b \in \mathbf{B}$. To fulfill this and to minimize the cardinality of ω, it was shown for linear static models in Chapter 3 that ω should be a subset of all feasible MO sets in the diagnosis model $\mathbb{M}$.

In a general diagnosis model not all MO subsets M of $\boldsymbol{M}$ are feasible models. For example in the model (3.95), the equation assumptions of 4 and 8 are mutually exclusive, i.e.,

$$\text{assump}(4) \cap \text{assump}(8) \neq \emptyset$$

Hence any subset that include both 4 and 8 will not be a feasible set, for example the MO set $\{4, 8\}$.

If we want to find all feasible MO sets in the set $\boldsymbol{M}$ of equations (3.95), one straightforward approach is to use $\{\{e\} | e \in \boldsymbol{M}\}$ as inputs to Algorithm 4. Since the algorithm does not consider the equation assumptions, non feasible MO sets will also be included in the output. All non feasible MO sets must therefore be removed from the output set. As said in Section 4.5, Algorithm 4 is efficient when the redundancy is low. To find all feasible MO sets in this way is inefficient because the redundancy of $\boldsymbol{M}$ can be much greater than the maximum redundancy of the maximal feasible models in $\boldsymbol{M}$. For the model (3.95), the redundancy of both maximal feasible models is 3 and the redundancy of $\boldsymbol{M}$ is 4. The redundancy difference will be greater for diagnosis models including more fault models.

To keep the redundancy of the models processed by Algorithm 4 low, another approach, that will be described here, is to call Algorithm 4 several times with different parts of the model as inputs. This approach is implemented in, Algorithm 7, for finding all feasible MO sets in a diagnosis model. Algorithm 7 computes MO sets by calling Algorithm 4 several times with different inputs. The set of all MO sets found by Algorithm 7 is the union of all output sets obtained for the different calls of Algorithm 4. Algorithm 7 uses the equation assumptions to compute inputs sets to Algorithm 4 such that only feasible MO sets are contained in the outputs of Algorithm 4. The algorithm presented here is not limited to linear models if Algorithm 4 is replaced by an algorithm handling also non-linear models. This will later be done in Chapter 9.

In Section 5.1 the relation between the inputs and the output of Algorithm 4 is investigated. This is then used in Section 5.2 to derive requirements on the inputs such that each feasible MO sets are found in exactly one of the calls of Algorithm 4. Many sets of inputs fulfill these requirements, but we are also interested to minimize the number of calls to Algorithm 4, and this is done in Section 5.3. The final algorithm is presented in Section 5.4 and Section 5.5. Finally the computational complexity of the algorithm is discussed in Section 5.6 before the conclusions are drawn in Section 5.7.

5.1 Relation between Inputs and Outputs of Algorithm 4

To keep the redundancy of the models processed by Algorithm 4 low, one approach is to call Algorithm 4 with all maximal feasible sets and then take the union of the output sets. All MO sets found using this approach will be feasible sets, but the drawback is that a node in the search tree corresponding to PO set M and its subtree will be traversed in each call to Algorithm 4 with a maximal feasible set M' if $M \subseteq M'$. For (3.95) this would mean that nodes in the subtree defined by $\{1, 2, 3, 5, 6, 7\}$, i.e., the intersection of the two maximal feasible models, would be searched twice.

One way to prohibit this is to check if a node has been visited before entering it. This can be done by checking if the computed PO set is among the previously considered PO sets. Since the redundancy is known only PO sets with the same redundancy must be compared. Although comparisons to only PO sets of the same redundancy is sufficient, the number of comparisons will be large.

Another way to prohibit that nodes are visited twice or more without explicitly checking if they have been visited before is to make a clever choice of the input sets to Algorithm 4 such that different sets of nodes are visited in each call. In particular, disjoint families of MO sets will be found in each call.

The disjoint families of MO sets can be specified by changing the inputs to Algorithm 4. In general, it is possible to specify the k:th inputs of Algorithm 4 according to a partition

$$\mathbf{M} = M_n^k \cup M_r^k \cup M_a^k \tag{5.1}$$

such that the output set $\mathcal{M}_{MO}$ only contains all MO sets $M \subseteq \mathbf{M}$ with the following properties:

a) no equations in $M_n^k \subseteq \mathbf{M}$ are included in M,

b) all equations in $M_r^k \subseteq \mathbf{M}$ are included in M, and

c) it is ambiguous if the equations in $M_a^k \subseteq \mathbf{M}$ are included in M or not.

Given such partition, the corresponding MO sets are obtained in the output set of Algorithm 4 by choosing the input sets according to the next theorem.

Theorem 5.1. *Let* $\boldsymbol{M}$ *be a set of equations partitioned as*

$$\boldsymbol{M} = M_n^k \cup M_r^k \cup M_a^k \tag{5.2}$$

If

$$M_r^k \subseteq (M_r^k \cup M_a^k)^* \tag{5.3}$$

then by choosing the input sets to Algorithm 4 as

$$\mathcal{S} = \{\{e\}|e \in (M_r^k \cup M_a^k)^*\} \tag{5.4}$$

and

$$\mathcal{E} = \{\{e\}|e \in (M_r^k \cup M_a^k)^* \cap M_a^k\} \tag{5.5}$$

the output set $\mathcal{M}_{MO}$ *will include exactly those MO sets in* $\boldsymbol{M}$ *that fulfill all the properties (a)-(c). Otherwise, i.e., if* (5.3) *is not true, no MO sets in* $\boldsymbol{M}$ *fulfill all the properties (a)-(c).*

Proof. First we start to prove that if (5.3) is not true, then no MO sets in $\boldsymbol{M}$ fulfill all the properties (a)-(c). For an MO set M satisfying property (a), it holds that $M \subseteq M_r^k \cup M_a^k$. Since all MO sets are contained in the overdetermined part according to Lemma 4.3, it follows that

$$M \subseteq (M_r^k \cup M_a^k)^* \tag{5.6}$$

Now the MO set M satisfies (b), if and only if all equations in M_r^k are included in the overdetermined part $(M_r^k \cup M_a^k)^*$, i.e., if and only if (5.3) is satisfied. Hence if (5.3) is not true, then no MO sets in $\boldsymbol{M}$ fulfill all properties (a)-(c).

From now on it is assumed that (5.3) is fulfilled. We start to describe the output set $\mathcal{M}_{MO}$ of Algorithm 4 formally. According to the description of Algorithm 4, the input set $\mathcal{S}$ contains the equations considered and $\mathcal{E}$ contains the equations that are allowed to be removed . Then from (5.4), (5.5), and Theorem 4.5, the set $\mathcal{M}_{MO}$ is the set of all MO sets that are subsets of $(M_a^k \cup M_r^k)^*$ and can be obtained by only removing equations in M_a^k. This means that the set $\mathcal{M}_{MO}$ contains all MO sets M such that

$$M \subseteq (M_a^k \cup M_r^k)^* \tag{5.7}$$

and by also using (5.3) it follows that

$$M_r^k \subseteq M \tag{5.8}$$

Next, we show that an arbitrary MO set $M \in \mathcal{M}_{MO}$ fulfills (a)-(c). Property (a) is implied by (5.7) and (b) is implied by (5.8). Property (c) is always fulfilled. Hence any MO set $M \in \mathcal{M}_{MO}$ fulfills (a)-(c).

Finally, we show that any MO set with properties (a)-(c) will be included in $\mathcal{M}_{MO}$. Let M be an arbitrary MO set with properties (a)-(c). From property (b), condition (5.8) follows. Property (a) implies that $M \subseteq (M_a^k \cup M_r^k)$ and by using Lemma 4.3 that $M \subseteq (M_a^k \cup M_r^k)^*$. Hence also the first condition (5.7) of the characterization is fulfilled. In conclusion any MO set that fulfills (a)-(c) belongs to $\mathcal{M}_{MO}$. □

5.2 Constructing Partitions by Using the Model

The MO sets specified by a partition (5.2) and the conditions (a)-(c) can be found by applying Algorithm 4 to the input sets described in Theorem 5.1. Now, a natural question is how to define the partitions $M_r^k \cup M_a^k \cup M_n^k$ such that the following two requirements are met:

I) For each partition only feasible MO sets should satisfy the conditions (a)-(c), i.e., only feasible MO sets shall be found by Algorithm 4.

II) All feasible MO sets in $\boldsymbol{M}$ should satisfy the conditions (a)-(c) for exactly one of the chosen partitions, i.e., all feasible MO sets will be found and all feasible MO sets are found only once.

The assumption of the equations determine if a set is feasible or not. Therefore, the set of partitions will be computed by using the equation assumptions.

There are many sets of partitions that satisfy these two requirements. For example, these two requirements are satisfied by the set of all partitions $\boldsymbol{M} = M_n^k \cup M_r^k$ where $M_r^k \subseteq \boldsymbol{M}$ is a feasible set. However, this is a computationally inefficient choice, because each feasible set will then be separately checked if it is an MO set. For an efficient use of Algorithm 4, the sets M_a^k in the chosen partitions should, in contrast to the example where all M_a^k were empty, be chosen large. In this way the number of partitions needed to specify all feasible MO sets becomes smaller.

To do this, the partitions are constructed as follows. Let the equations $\boldsymbol{M}$ be divided into two sets $\boldsymbol{M} = M_{ne} \cup M_e$ such that it is sufficient to consider only the equations in M_e to determine if any set is a feasible model or not. The set M_e is chosen as a minimal set such that any set $M \subseteq \boldsymbol{M}$ is a feasible set if and only if the part in M_e is a feasible set, i.e., M_e is a minimal set such that

$$\forall M \subseteq \boldsymbol{M} : (M \text{ feasible } \Leftrightarrow M \cap M_e \text{ feasible}) \tag{5.9}$$

The set M_{ne} is obviously the complement set of M_e in $\boldsymbol{M}$. Then it is possible to chose the sets M_r^k equal to all feasible subsets of M_e and M_a^k equal to the set M_{ne} in all partitions to guarantee that all MO sets specified by (a)-(c) for each partition k are feasible MO sets. As said before, $M_a^k = M_{ne}$ should be large, i.e., M_e should be small. That is the reason for defining M_e to be a minimal set that satisfies (5.9). In fact there exists a unique minimal set M_e that satisfies (5.9) according to the next lemma.

Lemma 5.1. *There exists a unique minimal set M_e that satisfies* (5.9).

Proof. Assume that M_{e1} and M_{e2} are two minimal sets that fulfill (5.9). We will show that $M_e = M_{e1} \cap M_{e2}$ also fulfills (5.9) and then it follows from the minimality of M_{e1} and M_{e2} that $M = M_{e1} = M_{e2}$.

To show that $M_e = M_{e1} \cap M_{e2}$ fulfills (5.9), let $M \subseteq \boldsymbol{M}$ be an arbitrary set and consider the set $M \cap (M_{e1} \cap M_{e2})$. Since M_{e2} fulfills (5.9), it follows that $(M \cap M_{e1}) \cap M_{e2}$ is feasible if and only if $M \cap M_{e1}$ is a feasible set. From the fact that M_{e1} fulfills (5.9), we get that $M \cap M_{e1}$ is a feasible set if and only if M is a feasible set. Hence we have proven that $M_e = M_{e1} \cap M_{e2}$ fulfills (5.9). From the minimality of M_{e1} and M_{e2} it follows that $M = M_{e1} = M_{e2}$ which completes the proof. □

Next an example of the partition $\boldsymbol{M} = M_e \cup M_{ne}$ is shown.

Example 5.1
Consider the electrical circuit in Example 3.17 with the set $\boldsymbol{M}$ of equations in (3.95) . A set in (3.95) is not a feasible model if and only if the set contains both equation 4 and 8. Hence the minimal set that satisfies (5.9) is

$$M_e = \{4, 8\} \tag{5.10}$$

The complement set of M_e in $\boldsymbol{M} = \{1, \ldots, 8\}$ is

$$M_{ne} = \{1, 2, 3, 5, 6, 7\} \tag{5.11}$$

In the example, M_e is the minimal set of equations with conflicting assumptions. In general M_e is the union of all minimal sets of equations with conflicting assumptions. Section 5.3 describes how M_e and M_{ne} can be computed. In the continuation here, we will assume that M_e and M_{ne} are known sets.

Given the sets M_e and M_{ne} the idea is to apply Algorithm 4 one time for each feasible set $M \subset M_e$. How to generate all feasible subsets M of M_e is discussed in Section 5.4. Given a feasible set M, we conclude that by choosing $M_n^k = M_e \setminus M$, $M_r^k = M$, and $M_a^k = M_{ne}$ in Theorem 5.1 we get the desired MO sets. In the next theorem we show that these partitions fulfill both requirement (I) and (II).

Theorem 5.2. *Let $\mathbb{M}$ be a diagnosis model with a set $\boldsymbol{M}$ of equations divided into the sets M_e and M_{ne} where M_e is defined as the minimal set that satisfies* (5.9). *Given all partitions $\boldsymbol{M} = M_r^k \cup M_n^k \cup M_a^k$ for $k = 1, \ldots$ such that*

$$M_r^k \subset M_e \tag{5.12}$$

is a feasible set,

$$M_n^k = M_e \setminus M_r^k \tag{5.13}$$

and

$$M_a^k = M_{ne} \tag{5.14}$$

the requirements (I) and (II) are fulfilled.

Proof. We start to show that requirement (I) is fulfilled. Assume that M is an MO set, not necessarily a feasible set, that satisfies (a)-(c) for some partition k of the type (5.2). Since M satisfies (a)-(c) it follows that

$$M_r^k \subseteq M \subseteq M_r^k \cup M_a^k \tag{5.15}$$

By substitution of M_{ne} in $M_{ne} \cap M_e = \emptyset$ for M_a^k using (5.14) we get that $M_a^k \cap M_e = \emptyset$. This implies that

$$(M_r^k \cup M_a^k) \cap M_e = M_r^k \cap M_e \tag{5.16}$$

By taking the intersecting each of the three sets in (5.15) with M_e and by using (5.16), it follows that

$$M \cap M_e = M_r^k \cap M_e$$

This and (5.12) imply

$$M \cap M_e = M_r^k$$

Since $M_r^k = M \cap M_e$ is a feasible set, (5.9) implies that M is a feasible set. Since M was an arbitrarily chosen MO set, it follows that only feasible MO sets satisfy (a)-(c) for some partition k.

Now, we show that requirement (II) is satisfied. Let $M \subseteq \boldsymbol{M}$ be an arbitrary feasible MO set. This implies according to (5.9) that the subset $M \cap M_e$ is a feasible set. For each partition k, the sets M_r^k and M_n^k divide the set M_e according to (5.12) and (5.13). From this, (a), and (b), it follows that any MO set that fulfills (a)-(c) for partition k contains exactly the equations M_r^k among the equations in M_e. This implies that M satisfies (a)-(c) only if $M_n^k = M_e \setminus M$, $M_r^k = M \cap M_e$, and $M_a^k = M_{ne}$. Since $M_r^k = M \cap M_e$ is a feasible subset of M_e it follows that $M_n^k = M_e \setminus M$, $M_r^k = M \cap M_e$, and $M_a^k = M_{ne}$ is one of the partitions defined in (5.12)-(5.14). Hence all feasible MO sets satisfy (a)-(c) for exactly one partition. □

We conclude this section with an example that shows the partitions obtained given the sets M_e and M_{ne}. Furthermore the example shows how these partitions are used to define inputs to Algorithm 4 such that all feasible MO sets are found.

Example 5.2

Continuation of Example 5.1. The set M_e and M_{ne} were given in (5.10) and (5.11) respectively. The feasible subsets of M_e are

$$\mathcal{M}_{feas} = \{\emptyset, \{4\}, \{8\}\}$$

The three corresponding partitions $\boldsymbol{M} = M_r^k \cup M_n^k \cup M_a^k$ for $k \in \{1, 2, 3\}$ are

$$\begin{array}{lll} M_r^1 = \emptyset & M_n^1 = \{4, 8\} & M_a^1 = M_{ne} \\ M_r^2 = \{4\} & M_n^2 = \{8\} & M_a^2 = M_{ne} \\ M_r^3 = \{8\} & M_n^3 = \{4\} & M_a^3 = M_{ne} \end{array}$$

These three partitions satisfy requirement (I) and (II) according to Theorem 5.2. From Theorem 5.1 we get that the first partition corresponds to the inputs

$$\begin{array}{rcl} \mathcal{S}_1 & = & \{\{1\}, \{2\}, \{3\}, \{5\}, \{6\}, \{7\}\} \\ \mathcal{E}_1 & = & \{\{1\}, \{2\}, \{3\}, \{5\}, \{6\}, \{7\}\} \end{array}$$

to Algorithm 4, the second to

$$\begin{array}{rcl} \mathcal{S}_2 & = & \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}\} \\ \mathcal{E}_2 & = & \{\{1\}, \{2\}, \{3\}, \{5\}, \{6\}, \{7\}\} \end{array}$$

and the third to

$$\begin{array}{rcl} \mathcal{S}_3 & = & \{\{1\}, \{2\}, \{3\}, \{5\}, \{6\}, \{7\}, \{8\}\} \\ \mathcal{E}_3 & = & \{\{1\}, \{2\}, \{3\}, \{5\}, \{6\}, \{7\}\} \end{array}$$

If Algorithm 4 is applied to these inputs, all feasible MO sets can be computed as

$$\mathcal{M}_{MO} = \bigcup_{k \in \{1,2,3\}} \texttt{FindMO}(\mathcal{S}_k, \mathcal{E}_k) \tag{5.17}$$

according to Theorem 5.1. In Figure 3.1 these three calls find the four MO sets in the middle, the four leftmost MO sets, and the four rightmost MO sets, respectively.

This example illustrates that all MO sets are found and each MO set is found only ones if the input sets $\mathcal{S}_k$ and $\mathcal{E}_k$ of the different calls of Algorithm 4 are chosen in accordance with the result of Theorem 5.1.

5.3 Computing the set M_{ne}

In this section we show how to compute the sets M_e and M_{ne} given a diagnosis model $\mathbb{M}$. The next lemma characterize the set M_{ne} given the set $\mathcal{M}_{max}$ of all maximal feasible subsets in $\boldsymbol{M}$ and the characterization will then be used to compute the set M_{ne}.

Lemma 5.2. *It holds that*

$$M_{ne} = \cap_{\hat{M} \in \mathcal{M}_{max}} \hat{M} \tag{5.18}$$

Proof. Let the set defined on the right hand side of (5.18) be denoted $\bar{M}$. Instead of proving that $\bar{M} = M_{ne}$ we will prove the equivalent statement that

$$(\boldsymbol{M} \setminus \bar{M}) = M_e \tag{5.19}$$

We start to show that

$$(\boldsymbol{M} \setminus \bar{M}) \supseteq M_e \tag{5.20}$$

Since M_e is defined to be the minimal set that satisfies (5.9), it follows that (5.20) is equivalent with (5.9) where M_e is substituted by $\boldsymbol{M} \setminus \bar{M}$. Take an arbitrary $M \subseteq \boldsymbol{M}$. The right implication of (5.9) is trivially true. To prove the left implication of (5.9), assume that $M \cap (\boldsymbol{M} \setminus \bar{M})$ is a feasible set. Since $M \cap (\boldsymbol{M} \setminus \bar{M})$ is feasible there exist a $\hat{M} \in \mathcal{M}_{max}$ such that

$$M \cap (\boldsymbol{M} \setminus \bar{M}) \subseteq \hat{M} \tag{5.21}$$

From the construction of $\bar{M}$, it follows that

$$\bar{M} \subseteq \hat{M}$$

and therefore also that

$$M \cap \bar{M} \subseteq \hat{M} \tag{5.22}$$

Since M can be written as

$$M = (M \cap \bar{M}) \cup (M \cap (\boldsymbol{M} \setminus \bar{M})) \tag{5.23}$$

(5.21) and (5.22) imply that

$$M \subseteq \hat{M} \tag{5.24}$$

By using this and that $\hat{M}$ is a feasible set, it follows that M is a feasible set. Hence (5.20) is proved.

Next, we show that

$$(\boldsymbol{M} \setminus \bar{M}) \subseteq M_e \tag{5.25}$$

i.e., $M_e = (\boldsymbol{M} \setminus \bar{M})$ is the minimal set that satisfies (5.9). Let $e \in \boldsymbol{M}$ be an arbitrary equation such that $e \notin \bar{M}$. To show that $(\boldsymbol{M} \setminus \bar{M})$ is a minimal set that satisfies (5.9), it is sufficient to show that (5.9) with M_e substituted for $\boldsymbol{M} \setminus (\bar{M} \cup \{e\})$ does not satisfy (5.9). Since $e \notin \bar{M}$, it follows that there exists an $\hat{M} \in \mathcal{M}_{max}$ such that

$$e \notin \hat{M} \tag{5.26}$$

By choosing $M = \hat{M} \cup \{e\}$ in (5.9), we get that

$$(\hat{M} \cup \{e\}) \text{ feasible } \Leftrightarrow (\hat{M} \cup \{e\}) \cap (\boldsymbol{M} \setminus (\bar{M} \cup \{e\})) \text{ feasible}$$

On the left hand side we have $\hat{M} \cup \{e\}$ which is a not feasible set according to the fact that $\hat{M}$ is a maximal feasible set. On the right hand side, we have the set

$$(\hat{M} \cup \{e\}) \cap (\boldsymbol{M} \setminus (\bar{M} \cup \{e\})) \tag{5.27}$$

Since $e \notin (\boldsymbol{M} \setminus (\bar{M} \cup \{e\}))$ it follows that (5.27) is a subset of the feasible set $\hat{M}$. Hence the set in (5.27) is feasible. This concludes that (5.9) is not satisfied when M_e is substituted for $\boldsymbol{M} \setminus (\bar{M} \cup \{e\})$. Since $e \in \boldsymbol{M}$ was arbitrarily chosen such that $e \notin \bar{M}$, (5.25) follows. Finally, (5.20) and (5.25) imply (5.18) which concludes the proof. □

The component based modeling described in Section 2.1 will be used in the computation of M_{ne}. Let $c \in \mathbf{C}$ be all components described in a diagnosis model $\mathbb{M}$. Let $M_c \subseteq \boldsymbol{M}$ be the equations describing the behavior of component c, e.g. $M_B = \{4, 8\}$ in Example 3.17. Furthermore let $M_0 = \boldsymbol{M} \setminus (\cup_{c \in \mathbf{C}} M_c)$ be all other equations. Note that the equations in M_0 can without loss of generality be assumed to be always true. Furthermore assume that $\mathcal{M}_{max,c}$ is the set containing all maximal feasible subsets of M_c. Any maximal feasible set $\hat{M}$ in $\mathcal{M}_{max}$ can then be written as

$$\hat{M} = \cup_{c \in \mathbf{C}} \hat{M}_c \cup M_0 \tag{5.28}$$

where $\hat{M}_c \in \mathcal{M}_{max,c}$. Since M_c are disjoint sets, the intersection in (5.18) of all maximal feasible sets of the type (5.28) can be expressed as

$$M_{ne} = \cup_{c \in \mathbf{C}} (\cap_{\hat{M} \in \mathcal{M}_{max,c}} \hat{M}) \cup M_0 \tag{5.29}$$

Hence the set of all maximal feasible sets $\mathcal{M}_{max,c}$ for each model M_c describing component c must be determined to compute the set M_{ne} when using (5.29). Since the number of component behavioral modes in general is small, computation of all maximal feasible models can easily be done as follows.

Remember that $\mathbf{B}_c$ denotes that set of all component behavioral modes for component c. Given a component $c \in \mathbf{C}$ and a component behavioral mode $b \in \mathbf{B}_c$, let $M_{c,b}$ be a component behavioral model defined by

$$M_{c,b} = \{e \in M_c | \phi(c = b) \subseteq \text{assump}(e)\} \tag{5.30}$$

Then the set $\mathcal{M}_{max,c}$ of all maximal feasible models in M_c can be computed as the maximal sets among

$$\{M_{c,b} | b \in M_c\} \tag{5.31}$$

Next an example shows the computation of the set $\mathcal{M}_{max,c}$ of maximal feasible sets of a component c.

Example 5.3

Consider for example a sensor component c with the possible behavioral modes:

no-fault NF, constant bias fault B, and unknown fault UF. If the measurement is denoted by y, the measured signal is denoted by x, and the sensor fault is modeled by a fault signal f, then a model for the sensor is

$$\begin{array}{ll} & y = x + f \\ c = \text{NF} & f = 0 \\ c \in \{\text{NF}, \text{B}\} & \dot{f} = 0 \end{array} \tag{5.32}$$

The sets of valid equations for each component behavioral mode, i.e., the sets in (5.31), are $\{1, 2, 3\}$ for NF, $\{1, 3\}$ for B, and $\{1\}$ for UF. The only maximal sets for this component is therefore $\{1, 2, 3\}$, i.e., $\mathcal{M}_{max,c} = \{\{1, 2, 3\}\}$.

The computations of the partition $\boldsymbol{M} = M_{ne} \cup M_e$ is summarized in the following algorithm. The intersection in (5.29) for component c is in the algorithm denoted by $\bar{M}_c$.

Algorithm 5. `GetPartition(`$\boldsymbol{M}$`)`

```
% Computes the partition $\boldsymbol{M} = M_{ne} \cup M_e$.
for each component c do

    $\mathcal{M}_{max,c} := \emptyset$;

    for each component behavioral mode $b \in \mathcal{B}_c$ do

        if $M_{c,b}$ not subset of any $M \in \mathcal{M}_{max,c}$ then
            Remove all subsets $M$ of $M_{c,b}$ in $\mathcal{M}_{max,c}$;
            $\mathcal{M}_{max,c} := \mathcal{M}_{max,c} \cup \{M_{c,b}\}$;
        end if

    end for

    $\bar{M}_c := \cap_{M \in \mathcal{M}_{max,c}} M$; % See (5.29)

end for
$M_0 := \boldsymbol{M} \setminus (\cup_{c \in \mathbf{C}} M_c)$;
$M_{ne} := (\cup_{c \in \mathbf{C}} \bar{M}_c) \cup M_0$;
$M_e := \boldsymbol{M} \setminus M_{ne}$;
return $M_{ne}$ and $M_e$;
```

Note that the sets $\mathcal{M}_{max,c}$ for $c \in \mathbf{C}$ could all be replaced by a single set reducing the memory usage.

Example 5.4

Consider the electrical circuit in Example 3.17. Assume that the components are enumerated in the order $R1$, $R2$, B, $S1$, $S2$, and $S3$. Considering the model (3.95), the sets M_{c_i} are

$$M_{c_i} = \begin{cases} \{i + 1\} & \text{for all } i \in \{0, 1, 2, 4, 5, 6\} \\ \{4, 8\} & \text{for } i = 3 \end{cases}$$

All components except for the battery have the possible component behavioral modes NF and UF. These component models have one maximal feasible model and this is the no-fault model, i.e., $\mathcal{M}_{max,c_i} = \{M_{c_i}\}$ for all $i \in \{1, 2, 4, 5, 6\}$. This implies that $\bar{M}_{c_i} = M_{c_i}$ for all $i \in \{1, 2, 4, 5, 6\}$ according to the algorithm.

The battery model $M_B = M_{c_3}$ consists of the equations 4 and 8 in (3.95). The battery has two possible component behavioral modes: NF with model {4}, and F with model {8}. Therefore, the maximal feasible models for component B is $\mathcal{M}_{max,c_3} = \{\{4\}, \{8\}\}$ and $\bar{M}_{c_3} = \emptyset$.

By substitution of the sets $\bar{M}_{c_i}$ and M_0 in the assignment of the set M_{ne} in Algorithm 5, the result becomes

$$M_{ne} = \{1, 2, 3, 5, 6, 7\} \tag{5.33}$$

The set M_e becomes

$$M_e = \{4, 8\}$$

which also is in agreement with the computations in Example 5.2.

5.4 Generating Feasible Subsets

In this section we present an algorithm that finds all feasible models $M \subseteq M_e$. A property that will be used to find these is that if a model is not feasible then no supersets will be a feasible model. Here the feasible models will be generated by starting with the empty set and then generate supersets following a depth first traversal. If a set turns out to be a non-feasible model then the corresponding branch is cut off.

Another feature that will be used is the component based modeling approach. Remember that M_c denotes the set of equations describing the behavior of component c. All feasible subsets of $M_c \cap M_e$ are first computed for each component $c \in \mathbf{C}$. Assume that the set of all feasible subsets of $M_c \cap M_e$ is denoted $\mathcal{M}_{feas,c}$. Then it follows that any feasible subsets of M_e can be written as

$$\cup_{c \in \mathbf{C}} \tilde{M}_c \tag{5.34}$$

for some $\tilde{M}_c \in \mathcal{M}_{feas,c}$.

The algorithm is a recursive algorithm for generating all feasible models contained in a general set U of equations. Later in the algorithm for finding all feasible MO sets U will be the sets $M_c \cap M_e$ where $c \in \mathbf{C}$. In each recursive call the algorithm returns given two input sets $M \subseteq U$ and $E \subseteq U$, the set $\mathcal{M}_{feas}$ containing all feasible sets that can be written as $M \cup E'$ where $E' \subseteq E$. This means especially that if $M = \emptyset$ and $E = U$, then the algorithm will return the set $\mathcal{M}_{feas}$ containing all feasible models in U.

Algorithm 6. `GetAllFeasibleSubsets`(M, E)

```
***if*** *M not feasible* ***then***

    $\mathcal{M}_{feas} := \emptyset$;

***else***

    $\mathcal{M}_{feas} := \{M\}$; $M_{old} := M$;

    ***while*** $E \neq \emptyset$ ***do***
```

```
        Select an $e \in E$;
        $E := E \setminus \{e\}$;
        $M := M_{old} \cup \{e\}$;
        $\mathcal{M}_{feas} := \mathcal{M}_{feas} \cup$ GetAllFeasibleSubsets$(M, E)$;
    end while
end if
return $\mathcal{M}_{feas}$;
```

Example 5.5
Continuing Example 5.4. From (5.33) we get that the complement set of $\boldsymbol{M}$ is $M_e = \{4, 8\}$. If Algorithm 6 is applied to the input $M_e \cap M_B = M_e \cap M_{c_3} = \{4, 8\}$, the output set becomes

$$\mathcal{M}_{feas} = \{\emptyset, \{4\}, \{8\}\}$$

5.5 Algorithm for Finding All Feasible MO Sets

Now, we are ready to include all parts in a final algorithm for finding all feasible MO sets in a diagnosis model $\mathbb{M}$ with an equation set $\boldsymbol{M}$.

The main steps in the algorithm is as follows. First the set $\boldsymbol{M}$ is divided into the sets $\boldsymbol{M} = M_{ne} \cup M_e$ with Algorithm 5 presented in Section 5.3. Then all feasible subsets $\mathcal{M}_{feas,c}$ of each component model $M_c \cap M_e$ are computed using Algorithm 6 described in Section 5.4. Given all feasible subsets $\mathcal{M}_{feas,c}$ of all components $c \in \mathbf{C}$, all feasible subsets M_r^k of M_e can be generated as

$$M_r^k = \cup_{c \in \mathbf{C}} \bar{M}_c \tag{5.35}$$

for all combinations of $\bar{M}_c \in \mathcal{M}_{feas,c}$ for all $c \in \mathbf{C}$. Then for each feasible set M_r^k, all MO sets specified by (a)-(c) for the partition M_r^k, $M_a^k = M_{ne}$, and $M_n^k = \boldsymbol{M} \setminus (M_r^k \cup M_a^k)$ are found by Algorithm 4.

The number of partitions M_r^k, M_a^k, M_n^k can be large. To avoid storing all partitions M_r^k, M_a^k, M_n^k, the algorithm recursively construct each partition M_r^k, M_a^k, M_n^k, apply Algorithm 4 to the constructed partition, and store the found MO sets before computing the next partition. The recursions are performed using a subroutine called FindFeasibleMOSets.

In the FindFeasibleMOSets the sets M_r^k is obtained as follows. Starting with $M_r = \emptyset$ the subroutine chose one feasible set of a component not previously considered in each call. In each call the union of the chosen feasible sets are stored as the set M_r. The final set M_r, which corresponds to one M_r^k, is obtained with $|\mathbf{C}|$ recursive calls. Then all feasible MO sets containing M_r and contained in $M_r \cup M_{ne}$ are found by using Algorithm 4.

Assume that a model $\bar{M}_c \in \mathcal{M}_{feas,c}$ for all components in $\mathbf{C} \setminus \mathbf{C}'$ is selected and that

$$M_r = \cup_{c \in \mathbf{C} \setminus \mathbf{C}'} \bar{M}_c \tag{5.36}$$

The inputs to the subroutine are the union M_r of previously selected feasible sets for components $\mathbf{C} \setminus \mathbf{C}'$, a set $\mathbb{M}_{feas} = \{\mathcal{M}_{feas,c} | c \in \mathbf{C}' \subseteq \mathbf{C}\}$ of selection choices of the remaining components $\mathbf{C}'$, and the set M_{ne}. The output is the set of all MO sets that can be written as

$$M_r \cup (\cup_{c \in \mathbf{C}'} E_c) \cup E$$

where $E_c \in \mathcal{M}_{feas,c}$ and $E \subseteq M_{ne}$. All feasible MO sets in $\boldsymbol{M}$ is therefore obtained in the output set for the input sets $M = \emptyset$, $\mathbb{M}_{feas} = \{\mathcal{M}_{feas,c} | c \in \mathbf{C}\}$, and M_{ne}. The algorithm is summarized next.

Algorithm 7. `FindAllFeasibleMOSets`($\boldsymbol{M}$)

```
% Computes the partition  $\boldsymbol{M} = M_{ne} \cup M_e$.
$[M_{ne}, M_e]$ := GetPartition($\boldsymbol{M}$);

% Computes the feasible sets for each component.
for each component $c \in \mathbf{C}$ do

    $\mathcal{M}_{feas,c}$ := GetAllFeasibleSubset($\emptyset, M_c \cap M_e$);

end for

% Computes all feasible MO sets.
$\mathbb{M}_{feas} := \{\mathcal{M}_{feas,c} | c \in \mathbf{C}\}$;
$\mathcal{M}_{MO}$ := FindFeasibleMOSets($\emptyset, \mathbb{M}_{feas}, M_{ne}$);
return $\mathcal{M}_{MO}$

FindFeasibleMOSets($M, \mathbb{M}_{feas}, M_{ne}$)
if $\mathbb{M}_{feas} = \emptyset$ then
% A feasible set $M \subset M_e$ has been found.
% Check the partition of $\boldsymbol{M}$ defined by $M$.

    if $\boldsymbol{M} \subseteq (M \cup M_{ne})^*$ then  % See (5.3)

        $\mathcal{S} = \{\{e\} | e \in (M \cup M_{ne})^*\}$;

        $\mathcal{E} = \{\{e\} | e \in (M \cup M_{ne}M)^* \cap M_{ne}\}$;

        $\mathcal{M}_{MO}$ := FindMO($\mathcal{S}, \mathcal{E}$);

    else

        $\mathcal{M}_{MO} := \emptyset$;

    end if

else
% Extend $M$ with feasible sets in $\mathbb{M}_{feas}$.

    $\mathcal{M}_{MO} := \emptyset$;

    $M_{old} := M$;

    Select an $\mathcal{M}_{select} \in \mathbb{M}_{feas}$;
```

```
        $\mathbb{M}_{feas} := \mathbb{M}_{feas} \setminus \{\mathcal{M}_{select}\}$;
        while $\mathcal{M}_{select} \neq \emptyset$ do
            Select an $E \in \mathcal{M}_{select}$;
            $\mathcal{M}_{select} := \mathcal{M}_{select} \setminus \{E\}$;
            $M := M_{old} \cup \{E\}$;
            $\mathcal{M}_{MO} := \mathcal{M}_{MO} \cup$ FindFeasibleMOSets($M, \mathbb{M}_{feas}, M_{ne}$);
        end while
end if
return $\mathcal{M}_{MO}$;
```

The algorithm is justified by the following result.

Theorem 5.3. *If Algorithm 7 is applied to a diagnosis model $\mathbb{M}$ with equations $\boldsymbol{M}$, then only feasible MO sets are found and any feasible MO set contained in $\boldsymbol{M}$ is found once and only once.*

Proof. Algorithm 5 computes a partition $\boldsymbol{M} = M_e \cup M_{ne}$ where M_e is the minimal set that satisfies (5.9) according to Lemma 5.2. Given M_e and M_{ne}, Algorithm 7 computes all feasible sets $M \subseteq M_e$ defining the partitions $\boldsymbol{M} = M_r^k \cup M_n^k \cup M_a^k$ where $M_r^k = M$, $M_n^k = M_e \setminus M$, and $M_a^k = M_{ne}$. According to Theorem 5.2 these partitions specify only feasible MO sets and each feasible MO sets is specified only for one partition. For each such partition, exactly the specified MO sets are then found by Algorithm 4 according to Theorem 5.1 and this concludes the proof. □

5.6 Computational Complexity

Besides the computational complexity of Algorithm 4, the number of feasible subsets $M \subseteq M_e$ in the for-loop in Algorithm 7 is important for the computational complexity of Algorithm 7. Consider an example model $\boldsymbol{M}$ with n components, i.e., $|\mathbf{C}| = n$. Assume that each component $c \in \mathbf{C}$ is specified by two equations $M_c = \{e_{c_1}, e_{c_2}\}$ with conflicting assumptions. The minimal conflicting set is $\{e_{c_1}, e_{c_2}\}$ and this set is equal to $M_e \cap M_c$. This implies that there are 3 feasible subsets of $M_e \cap M_c$, i.e., $|\mathcal{M}_{feas,c}| = 3$. This means that there are 3^n feasible subsets $M \subset M_e$. Even if each of these 3^n search problems are small only enumerating all cases will be time-consuming for large numbers n. However the next example shows that given a component c the number feasible subsets of $M_e \cap M_c$ might not increase with the number of component behavioral modes of c.

Example 5.6
In Example 5.3 there were 3 component behavioral modes. The set of all equation assumptions are consistent, i.e., $M_e \cap M_c = \emptyset$. The only feasible subset of $M_e \cap M_c$ is $\emptyset$, i.e., $\mathcal{M}_{feas,c} = \{\emptyset\}$. Hence, this component will not increase the number of cases in the while-loop in Algorithm 7.

5.7 Conclusions

An algorithm for computing all feasible MO sets of equations has been developed. There are three main ideas that are used in the algorithm. First, the algorithm uses Algorithm 4 several times and collects all MO sets found. Second, the inputs to Algorithm 4 are computed such that only feasible MO sets are contained in the outputs of Algorithm 4. Third and last, it is prohibited that any feasible MO set is found in more than one call of Algorithm 4. The algorithm presented here is not limited to linear models if Algorithm 4 is replaced by an algorithm handling also non-linear models.

6

S D L D S

Following the strategy described in Section 2.3.3, the construction of a diagnosis system starts with finding a set $\omega = \{M_1, \ldots, M_n\}$ of rejectable models to test. If the diagnosis system should be sound, these models $M_i \in \omega$ must fulfill

$$\mathcal{O}(M_b) = \bigcap_{M_i \subseteq M_b} \mathcal{O}(M_i) \tag{6.1}$$

for all $b \in \mathbf{B}$ according to Theorem 2.2. In Chapter 3, we studied linear static models and showed that the set ω of all feasible MO sets is a solution to (6.1) for all behavioral modes $b \in \mathbf{B}$. An algorithm for finding all feasible MO sets was then developed in Chapter 4 and Chapter 5. In this chapter, we will extend results presented in Chapter 3 to systems of linear differential equations. This is done by representing linear differential equations using polynomial matrices.

Basic properties of polynomial matrices are, for the sake of convenience, collected in Appendix 6.A. In Section 6.1 the type of differential equation considered in this chapter is specified along with some basic notions. Furthermore, the so called behavior of differential equations is defined and a polynomial matrix characterization of equivalent differential equations is described. By using the polynomial matrix characterization, observation set, redundancy, and overdetermined sets are extended to the dynamic case. Section 6.2 and Section 6.3 investigate solutions to (6.1) for one behavioral mode $b \in \mathbf{B}$. Section 6.2 extends the definition of PO set to the dynamic case and it is shown that all PO sets are sufficient for soundness. Section 6.3 introduces MO sets for dynamic models and shows that all MO sets are sufficient for soundness. In Section 6.4, it is shown that the set of all feasible MO sets satisfies (6.1) for all behavioral mode $b \in \mathbf{B}$. In Section 6.5, it is shown that Algorithm 7, presented in Chapter 5, can be used to find all feasible MO sets in a linear dynamic diagnosis model. Section 6.6 discusses how to find a minimum cardinality solution ω of MO sets. Finally, the fault influences on residual generators based on an MO sets are discussed in 6.7 before the conclusions are drawn in Section 6.8.

6.1 Linear Differential Equations

Let the set of all distributions be denoted by $\mathcal{D}'$. The type of model considered in this chapter is then a linear dynamic model for a specific behavioral mode

$$H(p)x + L(p)z = 0 \tag{6.2}$$

where $x \in \mathcal{D}'$ is an unknown distribution, $z \in \mathcal{D}'$ is a known distribution, $H(p) \in \mathbb{R}^{r \times n_x}[p]$ and $L(p) \in \mathbb{R}^{r \times n_z}[p]$ are known matrices with polynomial entries of the differential operator p. In this chapter x and z are assumed to be distributions, but all results hold also analogously if x and z instead are infinitely differentiable functions.

Throughout this chapter we will use the following example to illustrate concepts and theoretical results. Consider the linear dynamic model M_b:

$$\overbrace{\begin{bmatrix} p-1 \\ 1 \\ p \end{bmatrix}}^{H(p)} x + \overbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{L(p)} z = 0 \tag{6.3}$$

describing a fault free behavior.

6.1.1 Behavior

A key property that will be used for analyzing the solutions of linear differential equations is how the polynomial matrices are related for equivalent differential equations. This relationship is given in (Polderman and Willems, 1998) and here a part of this work is briefly recapitulated and exemplified.

The *behavior* of a linear differential equation $R(p)w = 0$ is defined as the set of all distributions $w \in \mathcal{D}'$ that are a solution to $R(p)w = 0$, i.e.,

$$\{w \in \mathcal{D}' | R(p)w = 0\} \tag{6.4}$$

The linear differential equations

$$R_1(p)w = 0 \quad R_2(p)w = 0 \tag{6.5}$$

are said to be *equivalent* if they define the same behavior. The next theorem characterize how the polynomial matrices $R_1(s)$ and $R_2(s)$ must be related to represent equivalent differential equations. The following theorem is given and proved in (Polderman and Willems, 1998).

Theorem 6.1 ((Polderman and Willems, 1998)). *Two polynomial matrices $R_1(s) \in \mathbb{R}^{p_1 \times q}[s]$ and $R_2(s) \in \mathbb{R}^{p_2 \times q}[s]$ where $p_1 \geq p_2$ define equivalent differential equations if and only if there exists a unimodular matrix $U(s)$ such that*

$$R_1(s) = U(s) \begin{bmatrix} R_2(s) \\ 0 \end{bmatrix} \tag{6.6}$$

Note that (6.6) is equivalent to that there exists a unimodular matrix $W(s)$ such that

$$W(s)R_1(s) = \begin{bmatrix} R_2(s) \\ 0 \end{bmatrix} \tag{6.7}$$

Multiplying the matrices of a differential equation from the left by any unimodular matrix will produce a differential equation equivalent to the original one according to Theorem 6.1. Unimodular matrices can be obtained by elementary row operations (see Appendix 6.A).

Example 6.1

To exemplify Theorem 6.1 and how elementary row operations can be used to obtain unimodular matrices consider the model (6.3). The rank of $H(s)$ is one and it is possible by elementary row operations to obtain only one non-zero row as

$$\begin{bmatrix} s-1 \\ 1 \\ s \end{bmatrix} \sim \begin{bmatrix} 1 \\ s-1 \\ s \end{bmatrix} \sim \begin{bmatrix} 1 \\ 0 \\ s \end{bmatrix} \sim \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{6.8}$$

by first interchange row 1 and 2, then multiply row 1 with $-(s-1)$ and add it to row 2, and finally multiply row 1 with $-s$ and add it to row 3. The elementary matrices corresponding to the elementary operations in (6.8) and the resulting unimodular matrix are

$$\begin{bmatrix} 1 & 0 & 0 \\ -s & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -(s-1) & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -(s-1) & 0 \\ 0 & -s & 1 \end{bmatrix} = U(s) \tag{6.9}$$

Since the elementary matrices are unimodular and the product of unimodular matrices is unimodular, it follows that $U(s)$ is a unimodular matrix. From the fact that $U(s)$ is unimodular, we get that

$$U(p)(H(p)x + L(p)z) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x + \begin{bmatrix} 0 & 1 & 0 \\ 1 & -(p-1) & 0 \\ 0 & -p & 1 \end{bmatrix} z = 0 \tag{6.10}$$

and (6.3) are equivalent differential equations according to Theorem 6.1.

6.1.2 Observation Sets

A dynamic model of the type (6.2) is said to be consistent with an observation, i.e., a distribution $z = z_0 \in \mathcal{D}'$ on any open time-interval $I \subset \mathbb{R}$, if

$$\exists x \in \mathcal{D}'; H(p)x + L(p)z_0 = 0 \tag{6.11}$$

For the model (6.11), the observation $[z_1(t), z_2(t), z_3(t)] = [\delta(t) - h(t), h(t), \delta(t)]$ where $\delta(t)$ is Dirac's delta function and $h(t)$ is Heaviside's step function is consistent with (6.2) because $x = h(t) \in \mathcal{D}'$ is then a solution to (6.3).

Let the equations in (6.2) be indexed by the set M. The set of all observations that can be observed when (6.11) holds is called the *observation set* of M and can formally be defined as

$$\mathcal{O}(M) = \{z \in \mathcal{D}' | \exists x \in \mathcal{D}'; H(p)x + L(p)z = 0\} \tag{6.12}$$

This means that a differential equation (6.2) is consistent with an observation, i.e, a distribution z, if the observation z belongs to the observation set $\mathcal{O}(M)$, i.e., if

$$z \in \mathcal{O}(M) \tag{6.13}$$

6.1.3 Kernel Representation

The observation set $\mathcal{O}(M)$ can compactly be expressed as the solution to a linear differential equation in z only, and this type of representation is called *kernel representation* (Polderman and Willems, 1998). In this section, it is shown that for any model there exists a kernel representation of the consistent observations. Furthermore a method to compute kernel representations is given. The existence of a kernel representation is first motivated by an example and then a general theorem is given.

Example 6.2
The last two equations in (6.10) is a kernel representation of the observation set of (6.3) and this can be understood as follows. Note that the two last equations, are equations only in the variable z. Furthermore the non-zero rows in $U(s)H(s)$ has full row rank and therefore will not impose any additional constraints of the solutions z. The set of consistent observations is therefore equal to the solutions of the two last equations only, i.e., to

$$\begin{bmatrix} 1 & -(p-1) & 0 \\ 0 & -p & 1 \end{bmatrix} z = 0 \tag{6.14}$$

This is therefore a kernel representation of the observation set of (6.10) and therefore also of the equivalent differential equation (6.3).

This example showed how a kernel representation could be constructed. In the next theorem, the existence of a kernel representation for any model is shown by using the same construction as in the example. This theorem is given in (Polderman and Willems, 1998), but there for sufficiently smooth functions.

Theorem 6.2. *The observation set (6.12) can be expressed as the solution to a differential equation in only the variables z in the following way:*

$$\mathcal{O}(M) = \{z \in \mathcal{D}' | N_H(p)L(p)z = 0\} \tag{6.15}$$

where N_H is any irreducible basis for the left null space of $H(s)$.

Proof. Let $z_0 \in \mathcal{O}(M)$, i.e., there exists an $x_0 \in \mathcal{D}'$ such that

$$H(p)x_0 + L(p)z_0 = 0 \tag{6.16}$$

There exists a unimodular matrix $U(s)$ such that

$$U(s)H(s) = \begin{bmatrix} U_1(s) \\ N_H(s) \end{bmatrix} H(s) = \begin{bmatrix} H_1(s) \\ 0 \end{bmatrix} \tag{6.17}$$

where $N_H(s)$ can be any irreducible basis for the left null space of $H(s)$ and $H_1(s)$ has full row rank. From Theorem 6.1 it follows that

$$U(p)(H(p)x_0 + L(p)z_0) = \begin{bmatrix} H_1(p) \\ 0 \end{bmatrix} x_0 + \begin{bmatrix} U_1(p)L(p) \\ N_H(p)L(p) \end{bmatrix} z_0 = 0 \tag{6.18}$$

and (6.16) are equivalent differential equations, i.e., the observation sets are equal. This implies that

$$N_H(p)L(p)z_0 = 0 \tag{6.19}$$

i.e.,

$$z_0 \in \{z \in \mathcal{D}' | N_H(p)L(p)z = 0\} \tag{6.20}$$

Since z_0 was arbitrarily chosen, it follows that

$$\mathcal{O}(M) \subseteq \{z \in \mathcal{D}' | N_H(p)L(p)z = 0\} \tag{6.21}$$

Now, let $N_H(s)$ be any irreducible basis of the left null space of $H(s)$ and take any solution $z_0 \in \{z \in \mathcal{D}' | N_H(p)L(p)z = 0\}$, i.e.,

$$N_H(p)L(p)z_0 = 0 \tag{6.22}$$

For any irreducible basis $N_H(s)$ of the left null space of $H(s)$, we can extend the matrix to a unimodular matrix $U(s)$ that satisfies (6.17). Then z_0 belongs to the set $\mathcal{O}(M)$ in (6.12) if and only if

$$\exists x \in \mathcal{D}'; \begin{bmatrix} H_1(p) \\ 0 \end{bmatrix} x + \begin{bmatrix} U_1(p)L(p) \\ N_H(p)L(p) \end{bmatrix} z_0 = 0$$

according to Theorem 6.1. Since (6.22) holds this is equivalent to that there exists an $x \in \mathcal{D}'$ such that

$$H_1(p)x + U_1(p)L(p)z_0 = 0 \tag{6.23}$$

From the fact that $H_1(p)$ has full row rank, it follows that for any distribution $U_1(p)L(p)z_0$ that there exists a distribution $x_0 \in \mathcal{D}'$ that satisfies (6.23). Hence $z_0 \in \mathcal{O}(M)$. Since z_0 was arbitrarily chosen, it follows that

$$\mathcal{O}(M) \supseteq \{z \in \mathcal{D}' | N_H(p)L(p)z = 0\} \tag{6.24}$$

The theorem follows by combining the subset relations in (6.21) and (6.24). □

The kernel representation (6.14) of the observation set of (6.3) was obtained by computing a unimodular matrix. However it is not necessary to compute a unimodular matrix to obtain an irreducible basis for the left null space of $H(s)$. A *left minimal basis in echelon form* can be directly be computed in an efficient and numerically stable way and there is an implementation in Matlab polynomial toolbox.

Example 6.3
The kernel representation of H in (6.3) can be computed as follows:

```
>> H=[s-1;1;s]

H =

    -1 + s
     1
     s

>> Nh = null(H')'

Nh =

     0.58      0.58 - 0.58s      0
    -0.58    -0.58               0.58
```

The matrix in (6.14) is one example of an irreducible basis for the left null space of $H(s)$ and the matrix `Nh` is another such example.

6.1.4 Redundancy and Overdetermined Sets

To generalize redundancy defined for static systems to dynamic systems, we consider polynomial matrices instead of real matrices. By using the *normal rank* (see Appendix 6.A) for polynomial matrices the redundancy concept is generalized to a set of linear differential equations as follows. The notation $H[M]$ will be extended to polynomial matrices and the dependence on s will for notational convenience only be explicit when needed.

Definition 6.1 (Redundancy of a Model). *Let $\varphi : 2^{\mathbf{M}} \to \mathbb{Z}$ be a function from the family of subsets M in a set $\mathbf{M}$ of linear differential equations (6.2) defined by*

$$\varphi(M) = \text{rank}([H[M]\, L[M]]) - \text{rank}(H[M]) \tag{6.25}$$

*This number $\varphi(M)$ will be called the **redundancy of** M.*

If the notation of the zero-padded matrix $N_{H[M]}$ given in Section 3.1.2 is extended to polynomial matrices, then the redundancy can also be written as

$$\varphi(M) = \text{rank}(N_{H[M]}L) \tag{6.26}$$

which can be interpreted as the maximum number of linearly independent consistency relations that can be derived from the set M of differential equations. For a linear dynamic model M, existence of analytical redundancy according to Definition 2.6 is equivalent to that $\varphi(M) > 0$. As for linear static models, dynamic linear models with redundancy is said to be *overdetermined* according to the next definition.

Definition 6.2 (Overdetermined Set). *A set M of linear differential equations (6.2) is an **overdetermined set** if its redundancy is positive, i.e., $\varphi(M) > 0$.*

Equation (6.26) and Theorem 6.2 imply that a linear dynamic model is an overdetermined set if and only if it is a rejectable model. Therefore, only overdetermined models need to be consider when choosing models ω in (6.1).

6.2 PO Sets are Sufficient for Soundness

In Section 3.2, it was shown for a linear static model M_b that it was sufficient to include any PO set $M \subseteq M_b$ with maximal redundancy in ω to fulfill (6.1). Surprisingly this is not true in general for the dynamic case. Here, we point out the differences and also show that if $[H(p)\, L(p)]$ has full row-rank then results analogous to the static case follow.

First we extend the definition of PO sets from linear static equations to linear differential equations.

Definition 6.3 (Proper Overdetermined Set). *An overdetermined set M of linear differential equations (6.2) is a **proper overdetermined (PO) set** if*

$$\varphi(E) < \varphi(M) \tag{6.27}$$

for all $E \subset M$.

The proper overdetermined sets in (6.3) are all subsets with two or three equations. If M is a proper overdetermined model, it follows that $\mathcal{O}(M) \subset \mathcal{O}(E)$ for all $E \subset M$. For the static case the converse implication is also true and shown in Theorem 3.1, but in the dynamic case it is not true as the next example shows.

Example 6.4
Let L be

$$L = \begin{bmatrix} s+2 & s+1 & 0 \\ s+3 & 0 & s+1 \\ 0 & s+3 & -(s+2) \end{bmatrix} \tag{6.28}$$

and let H be the empty matrix. We will now show that $\{1,2,3\}$ is not a PO set even if $\mathcal{O}(\{1,2,3\}) \subset \mathcal{O}(E)$ for any $E \subset \{1,2,3\}$. The redundancy for (6.28), when $H \in \mathbb{R}^{3\times 0}$, is

$$\varphi\,(M) = \text{rank}\,(L[M]) \tag{6.29}$$

The rank of L is 2 and the rank of two arbitrary rows $L[E]$ in L is also 2. Hence, $\{1,2,3\}$ is according to Definition 6.3 not a proper overdetermined set. However, all proper subsets E of $\{1,2,3\}$ fulfills

$$\mathcal{O}(\{1,2,3\}) \subset \mathcal{O}(E) \tag{6.30}$$

and it can be explained as follows. Sets with cardinality 1 has redundancy 1 and can obviously not have the same observation set as $\{1,2,3\}$. Therefore it is sufficient to study sets with cardinality 2, for example $E = \{1,2\}$. The matrix $L[E]$ spans the same rational space as L, but L has no roots and $L[E]$ has a root. This implies that there exists no unimodular matrix U that satisfies

$$L = U \begin{bmatrix} L[E] \\ 0 \end{bmatrix} \tag{6.31}$$

Then Theorem (6.6) implies that the observation sets for E and $\{1,2,3\}$ are different. The difference between the observation sets can be written as follows. An arbitrary trajectory of $\mathcal{O}(\{1,2\})$ can be written as

$$z + \alpha[0\,1\,0]^T e^{-t}$$

where z is a trajectory in $\mathcal{O}(\{1,2,3\})$ and α is a real constant. For any pair of rows in L, a similar argumentation holds.

In this example $[H(s)\,L(s)]$ had not full row-rank. Next it is shown that if $[H(s)\,L(s)]$ has full row-rank, then results analogous to results presented in Section 3.3 for the static case follow.

Theorem 6.3. *An overdetermined set M of linear differential equations in the form (6.2) where $[H\,L]$ has full row rank has a unique subset M^* that is a proper overdetermined set and fulfills*

$$\mathcal{O}(M) = \mathcal{O}(M^*) \tag{6.32}$$

Proof. The non-zero columns in N_H is uniquely determined and Theorem 4.2 states that these columns correspond to the equations in the set M^*. This means that $N_H L z$ is a kernel representation of both $\mathcal{O}(M)$ and $\mathcal{O}(M^*)$, which directly implies (6.32). □

This result implies that M is a proper overdetermined set if $\mathcal{O}(M) \subset \mathcal{O}(E)$ for all $E \subset M$ and it can be realized as follows. Assume that $\mathcal{O}(M) \subset \mathcal{O}(E)$ for all $E \subset M$. From this and that there exists a unique proper overdetermined set $M^* \subseteq M$ such that $\mathcal{O}(M^*) = \mathcal{O}(M)$, it follows that $M^* = M$, i.e., M is a proper overdetermined set. In the continuation we will always assume that $[H\,L]$ has full row-rank. It is always possible to find an equivalent differential equation described by $[H'\,L']$ such that $[H'\,L']$ has full row rank (Polderman and Willems, 1998). Therefore, we can without loss of generality assume that $[H\,L]$ has full row rank.

6.3 MO Sets are Sufficient for Soundness

A rank-condition given in Corollary 3.2 could be used to determine if a set of PO sets ω satisfies (6.1) for a linear static model. In this section a corresponding condition for linear dynamic models is developed. As in the static case, see Corollary 3.3, it is also shown that it is sufficient to include all MO sets in ω to satisfy (6.1).

6.3.1 Defining and Characterizing MO sets

The definition of MO sets for differential equations is analogous to the definition for static equations.

Definition 6.4 (Minimal Overdetermined Set). *An overdetermined set M of differential equations (6.2) is a* ***minimal overdetermined (MO) set*** *if no proper subset is an overdetermined set.*

The relation between MO sets and PO sets given in Lemma 3.2 for static equations holds also for the dynamic case.

Lemma 6.1. *A set M of equations (6.2) is an MO set if and only if M is a PO set with* $\varphi\,(M) = 1$.

Proof. The proof is analogous to the proof of Lemma 3.2. □

6.3.2 Rank Condition

In the linear static case we have given in Theorem 3.6 and Corollary 3.2 conditions for a set ω of models to fulfill (6.1). These conditions are not sufficient in the dynamic case and the complication is the irreducibility as the next example will show.

Example 6.5
The matrix $N_H L$ for the four PO sets in (6.3) are

$$\begin{array}{lll}
\text{PO set} & N_H L & \text{redundancy} \\
\hline
\{1,2,3\} & \begin{bmatrix} 1 & 1 & -1 \\ 0 & s & -1 \end{bmatrix} & \varphi\ = 2 \\
\{1,2\} & \begin{bmatrix} 1 & -(s-1) & 0 \end{bmatrix} & \varphi\ = 1 \\
\{1,3\} & \begin{bmatrix} s & 0 & -(s-1) \end{bmatrix} & \varphi\ = 1 \\
\{2,3\} & \begin{bmatrix} 0 & s & -1 \end{bmatrix} & \varphi\ = 1
\end{array} \tag{6.33}$$

The three last PO sets are MO sets, since their redundancies are one. Assume that we want to investigate if it is sufficient to check the consistency of $\{1,2,3\}$ by checking the consistency of any pair of MO sets. For a set of linear static equations with redundancy 2, two MO sets should be enough, according to Theorem 3.6. The rank-condition (3.50) is satisfied for all pairs of MO sets and therefore one would perhaps expect that any pair of MO sets can be used to check the consistency of $\{1,2,3\}$.

Assume that we want to investigate if the MO sets $\{1,2\}$ and $\{1,3\}$ can be used to check the consistency of $\{1,2,3\}$, i.e., if

$$\mathcal{O}(\{1,2,3\}) = \mathcal{O}(\{1,2\}) \cap \mathcal{O}(\{1,3\}) \tag{6.34}$$

Let P be

$$P = \begin{bmatrix} 1 & -(s-1) & 0 \\ s & 0 & -(s-1) \end{bmatrix} \tag{6.35}$$

where the first row in P corresponds to the MO set $\{1,2\}$ and the second row in P to $\{1,3\}$. Theorem 6.1 implies that an equivalent matrix formulation of (6.34) is that there exists a unimodular matrix U such that

$$\begin{bmatrix} 1 & 1 & -1 \\ 0 & s & -1 \end{bmatrix} = U \begin{bmatrix} 1 & -(s-1) & 0 \\ s & 0 & -(s-1) \end{bmatrix} \tag{6.36}$$

Analogously to the static case it follows that matrix P corresponding to the two MO sets has rank 2 and spans the same the space as the matrix $N_H L$ corresponding to the PO set $\{1,2,3\}$. Matrices spanning the same space describe equivalent systems in the static case but in the dynamic case equivalent differential equations are characterized by the stronger condition given in Theorem 6.1. The equations (6.36) is false, because $N_H L$ is irreducible but P has a root for $s = 1$. Hence it follows that

$$\mathcal{O}(\{1,2,3\}) \subset \mathcal{O}(\{1,2\}) \cap \mathcal{O}(\{1,3\}) \tag{6.37}$$

which means that it is not sufficient to check the consistency of $\{1,2\}$ and $\{1,3\}$ to determine if $\{1,2,3\}$ is consistent.

The same type of analysis reveals that the only pair of minimal overdetermined models that can be used to check the consistency of $\{1,2,3\}$ is $\{1,3\}$ and $\{2,3\}$. The equation corresponding to (6.36) is then

$$\begin{bmatrix} 1 & 1 & -1 \\ 0 & s & -1 \end{bmatrix} = U \begin{bmatrix} 1 & -(s-1) & 0 \\ 0 & s & -1 \end{bmatrix} \tag{6.38}$$

where U is

$$U = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{6.39}$$

In the example, the matrix defined by the MO sets $\{1,3\}$ and $\{2,3\}$ has no roots, in contrast to the pair of MO sets studied first. In the next theorem it is shown that the dynamic result corresponding to Corollary 3.2 is obtained by the additional requirement that the matrix $[N_{H[M_1]}^T \cdots N_{H[M_n]}^T]$ has no roots.

Theorem 6.4. *Let M be a PO set of linear differential equations (6.2) and let $M_i \subset M$ be PO sets for all $i \in I = \{1, 2, \ldots, n\}$ .Then it holds that*

$$\mathcal{O}(M) = \cap_{i \in I} \mathcal{O}(M_i) \tag{6.40}$$

if and only if

$$\operatorname{rank}\left(\begin{bmatrix} N^T_{H[M_1]} & N^T_{H[M_2]} & \cdots & N^T_{H[M_n]} \end{bmatrix}^T\right) = \operatorname{rank}(N_H) \tag{6.41}$$

and

$$\begin{bmatrix} N^T_{H[M_1]} & N^T_{H[M_2]} & \cdots & N^T_{H[M_n]} \end{bmatrix} \tag{6.42}$$

has no roots.

The result is discussed, before the proof is presented. The conditions (6.41) and (6.42) can be used to find out if it is possible to check the consistency of the models M_i to determine the consistency of M. Note that (6.41) is the same condition as (3.55) derived in the static case. Hence the only difference from the static case is to verify (6.42). This can be done efficiently by the function `roots` in matlab polynomial toolbox.

Example 6.6

Consider Example 6.5. Let P be the matrix obtained by the three rows $N_H L$ in (6.33) for each MO set. In this example, $N_H L = N_H$ because $L = I$. By the following commands in matlab, it is possible to find out that the first and second MO set do not correspond to an irreducible matrix but the first and third MO set do that.

```
>> P = [1 -(s-1) 0;s 0 -(s-1);0 s -1];
>> root = roots(P([1 2],:))

root =

   1.0000

>> root = roots(P([1 3],:))

root =

     []
```

To prove Theorem 6.4, two lemmas are introduced. In the formulation of Theorem 6.4, N_H is analyzed instead of the kernel representation in the form $N_H L$. In the next lemma we show that if $[H\, L]$ has full row rank then it is equivalent to study N_H and $N_H L$, i.e., L can be dropped in the dynamic case as well.

Lemma 6.2. *Let U, H, and L be polynomial matrices such that $[H\, L]$ has full row rank. Furthermore, if M is row-set of H and $M_1, \ldots, M_n$ are arbitrary subsets of M, then it follows that*

$$U \begin{bmatrix} N^T_{H[M_1]} & N^T_{H[M_2]} & \cdots & N^T_{H[M_n]} \end{bmatrix}^T = \begin{bmatrix} N_H \\ 0 \end{bmatrix} \tag{6.43}$$

if and only if

$$U\left[(N_{H[M_1]}L)^T \quad (N_{H[M_2]}L)^T \quad \cdots \quad (N_{H[M_n]}L)^T\right]^T = \begin{bmatrix} N_H L \\ 0 \end{bmatrix} \tag{6.44}$$

Proof. By multiplying L from the right in (6.43), we get (6.44).

To prove the if-part, assume that (6.44) is true. The matrices N_H and $N_{H[M_i]}$ belong to the left null space of H for any i and (6.44) can therefore be written as

$$U\left[(N_{H[M_1]}[H\,L])^T \quad (N_{H[M_2]}[H\,L])^T \quad \cdots \quad (N_{H[M_n]}[H\,L])^T\right]^T = \begin{bmatrix} N_H[H\,L] \\ 0 \end{bmatrix}$$

or equivalently,

$$(U\begin{bmatrix} N^T_{H[M_1]} & N^T_{H[M_2]} & \cdots & N^T_{H[M_n]} \end{bmatrix}^T - \begin{bmatrix} N_H \\ 0 \end{bmatrix})\begin{bmatrix} H & L \end{bmatrix} = 0 \tag{6.45}$$

Since $[H\,L]$ has full rank, i.e., the rows are linearly independent, it follows from (6.45) that (6.43) holds. □

To investigate if MO sets can be used to check the consistency of PO sets, we have seen that it corresponds to check if there exist a unimodular matrix that satisfies an equation of the type (6.36). Looking at Theorem 6.4 this condition is split into one rank condition (6.41) and one root condition (6.42). The next lemma states the equivalence between the existence of a unimodular matrix in (6.36) and the rank condition together with the root condition, which both easily can be checked.

Lemma 6.3. *Let $A(s)$ be an irreducible polynomial matrix with full row rank. A polynomial matrix $B(s)$ fulfills*

$$\text{rank}\,(\begin{bmatrix} A(s) \\ B(s) \end{bmatrix}) = \text{rank}\,(A(s)) \tag{6.46}$$

and

$$\text{rank}\,(A(s)) = \text{rank}\,(B(s_0)) \tag{6.47}$$

for all $s_0 \in \mathbb{C}$ if and only if there exists a unimodular matrix $U(s)$ such that

$$U(s)B(s) = \begin{bmatrix} A(s) \\ 0 \end{bmatrix} \tag{6.48}$$

Proof. Since $B(s)$ has no roots, it is possible to find a unimodular matrix $U'(s)$ such that

$$U'(s)B(s) = \begin{bmatrix} B'(s) \\ 0 \end{bmatrix}$$

where $B'(s)$ is irreducible. From (6.46) and (6.47), it follows that $A(s)$ and $B'(s)$ span the same space. Since both $A(s)$ and $B'(s)$ are irreducible, it follows from Theorem 6.A.3 that $B'(s) = W(s)A(s)$ and $A(s) = W'(s)B(s)$ where $W(s)$ and $W'(s)$ are square non-singular matrices. Combining these two expressions, we get

$$A(s) = W'(s)B(s) = W'(s)W(s)A(s)$$

Since $A(s)$ has full row rank, it follows that $W'(s)W(s) = I$, i.e., $W'(s) = W^{-1}(s)$ and both $W(s)$ and $W'(s)$ are unimodular matrices. Hence the matrix

$$U(s) = \begin{bmatrix} W'(s) & 0 \\ 0 & I \end{bmatrix} U'(s)$$

is a unimodular matrix that satisfies (6.48). Contrary if there exist a unimodular matrix $U(s)$ that satisfies (6.48), then (6.46) and (6.47) follow directly. □

Now, we are ready to prove Theorem 6.4.

Proof. The observation sets in (6.40) can according to Theorem 6.2 be written

$$\mathcal{O}(M) = \{z | N_H L z = 0\} \tag{6.49}$$

and

$$\cap_{i \in I} \mathcal{O}(M_i) = \{z | \forall i \in I; N_{H[M_i]} L z = 0\} \tag{6.50}$$

These sets are equal if and only if there exists a unimodular matrix U such that

$$U \begin{bmatrix} (N_{H[M_1]}L)^T & (N_{H[M_2]}L)^T & \cdots & (N_{H[M_n]}L)^T \end{bmatrix}^T = \begin{bmatrix} N_H L \\ 0 \end{bmatrix} \tag{6.51}$$

according to Theorem 6.1. The matrix $[H\, L]$ has full row rank, because M is a PO set. This implies according to Lemma 6.2 that (6.51) is equivalent to

$$U \begin{bmatrix} N_{H[M_1]}^T & N_{H[M_2]}^T & \cdots & N_{H[M_n]}^T \end{bmatrix}^T = \begin{bmatrix} N_H \\ 0 \end{bmatrix} \tag{6.52}$$

Since N_H is an irreducible basis, it fulfills the condition on $A(s)$ in Lemma 6.3 and this means that there exist a unimodular matrix U such that (6.52) holds if and only if

$$\text{rank}\left(\begin{bmatrix} N_{H[M_1]}^T(s_0) & N_{H[M_2]}^T(s_0) & \cdots & N_{H[M_n]}^T(s_0) \end{bmatrix}\right) = \text{rank}\,(N_H(s)) \tag{6.53}$$

for all $s_0 \in \mathbb{C}$ and

$$\text{rank}\left(\begin{bmatrix} N_H^T & N_{H[M_1]}^T & N_{H[M_2]}^T & \cdots & N_{H[M_n]}^T \end{bmatrix}\right) = \text{rank}\,(N_H) \tag{6.54}$$

Since $M_i \subset M$, (6.54) is trivially true. Furthermore (6.53) is equivalent to that both (6.41) and (6.42) are fulfilled. □

6.3.3 All MO Sets are Sufficient for Soundness

In (6.33) we saw that there existed only one pair of MO sets that could be used to check the consistency of $\{1, 2, 3\}$. In the next example will show that there are cases where all MO subsets of a PO set has to be checked to determine if the PO set is consistent. After that, we will prove that it is sufficient to check all MO sets if $[H\, L]$ has full row rank.

Example 6.7

Consider a model defined by

$$H = \begin{bmatrix} -(s+1) \\ s+2 \\ s+3 \end{bmatrix} \tag{6.55}$$

and $L = I$. The redundancy for this model is 2 and a basis for the null-spaces of the three MO sets are

$$P = \begin{bmatrix} N_{H[\{1,2\}]} \\ N_{H[\{1,3\}]} \\ N_{H[\{2,3\}]} \end{bmatrix} = \begin{bmatrix} s+2 & s+1 & 0 \\ s+3 & 0 & s+1 \\ 0 & s+3 & -(s+2) \end{bmatrix} \tag{6.56}$$

We see that $P[\{1,2\}]$ has a root $s = -1$, $P[\{1,3\}]$ has a root $s = -2$, and $P[\{2,3\}]$ has a root $s = -3$. Hence there exists no pair of MO sets that can be used to check the consistency of the PO set according to Theorem 6.4. However the matrix P has no roots which implies that the consistency of $\{1,2,3\}$ can be determined by checking the consistency of all MO sets.

Next, we will state and prove that the consistency of any PO set M can be determined by checking the consistency of all MO subsets. This result is the dynamical correspondence to the result in Theorem 3.7 given for static models.

Theorem 6.5. *If M is a PO set of linear differential equations, and E_i are all MO sets such that $E_i \subseteq M$, then it follows that*

$$\mathcal{O}(M) = \cap_i \mathcal{O}(E_i) \tag{6.57}$$

The proof is postponed to the end of this section. Combining the results from Theorem 6.3 and Theorem 6.5 we get the dynamic correspondence to Corollary 3.3.

Corollary 6.1. *Given a linear dynamic behavioral model M_b of the type (6.2) where $[H\,L]$ has full row-rank, it follows that*

$$\mathcal{O}(M_b) = \cap_{M_i \in \mathcal{M}_{MO}} \mathcal{O}(M_i) \tag{6.58}$$

where $\mathcal{M}_{MO}$ are the family of all MO sets $M \subseteq M_b$.

Proof. Since $[H\,L]$ has full row-rank, it follows that there exists a PO set $M^* \subseteq M_b$ such that

$$\mathcal{O}(M_b) = \mathcal{O}(M^*)$$

according to Theorem 6.3. Theorem 6.5 then implies that

$$\mathcal{O}(M_b) = \mathcal{O}(M^*) = \cap_{E' \in \mathcal{M}'_{MO}} \mathcal{O}(E')$$

where $\mathcal{M}'_{MO}$ are all MO sets that is a subset of M^*. Hence $\mathcal{M}'_{MO} \subseteq \mathcal{M}_{MO}$ and the corollary follows. □

If $[H\,L]$ has full row rank and ω is the set of all MO sets in a static or dynamic behavioral model M_b, then the set ω satisfies (6.1).

Example 6.8

To give an example, consider the model in (6.3). The matrix $[H\,L]$ has full row-rank and there are three MO sets, $\{1,2\}$, $\{1,3\}$, and $\{2,3\}$. By checking the consistency of these three sets the consistency of $\{1,2,3\}$ can be determined according to Theorem 6.5.

Finally, Theorem 6.5 is proved. The proof of this theorem uses the following lemma.

Lemma 6.4. *If M is a PO set with redundancy $\varphi > 1$ and $E_i \subset M$ where $i \in \{1, 2, \ldots, n\} = I$ are all PO sets with redundancy $\varphi - 1$, then it holds that*

$$\mathcal{O}(M) = \cap_{i \in I} \mathcal{O}(E_i) \tag{6.59}$$

Proof. Let N_i denote an irreducible basis for the left null space of H_i and let N denote an irreducible basis for the left null space of H. The equality (6.59) holds if and only if

$$\text{rank}\,(\begin{bmatrix} N_1^T(s) & N_2^T(s) & \cdots & N_n^T(s) \end{bmatrix}) = \text{rank}\,(N) \tag{6.60}$$

for all $s \in \mathbb{C}$ according to Theorem 6.4. All matrices $N_i(s)$ have normal rank $\varphi - 1$ and are irreducible, i.e., have full rank for all $s \in \mathbb{C}$. If M_i are the equivalence classes of M defined in Section 4.4.1, then

$$\begin{bmatrix} N_1(s) \\ N_2(s) \\ \vdots \\ N_n(s) \end{bmatrix} = \begin{bmatrix} 0 & N_1[:, M_2] & \cdots & N_1[:, M_n] \\ N_2[:, M_1] & 0 & \cdots & N_2[:, M_n] \\ \vdots & & \ddots & \vdots \\ N_n[:, M_1] & N_n[:, M_2] & \cdots & 0 \end{bmatrix} \tag{6.61}$$

where no column in $N_i[:, M_j]$ is zero for all $i \neq j$. Let the matrix in (6.61) be denoted by N_e. The rank of N_e is limited by

$$\varphi - 1 = \text{rank}\,(N_i) \leq \text{rank}\,(N_e) \leq \text{rank}\,(N) = \varphi$$

Furthermore, the matrix N_e has not rank $\varphi - 1$, because for example not all rows in N_2 are linearly dependent of N_1. This follows from the fact that $N_2[:, M_1] \neq 0$ and $N_1[:, M_1] = 0$. This implies that rank $(N_e) = \varphi$.

Now, it remains to prove that $N_e(s)$ has rank φ for all $s \in \mathbb{C}$. Assume that there exists an s_0 such that rank $(N_e(s_0)) < \varphi$. Since $N_i(s)$ are irreducible matrices with rank $(N_i(s)) = \varphi - 1$ for all $s \in \mathbb{C}$, it follows that rank $(N_e(s_0)) = \varphi - 1$. Then for an arbitrary $i \in I$, $N_i(s_0)$ is a basis for the space spanned by $N_e(s_0)$. Hence any row in $N_e(s_0)$ can be written as a linear combination of the rows in $N_i(s_0)$ for any $i \in I$. Let m be an arbitrary row of $N_e(s_0)$. Then

$$m = \gamma_i N_i(s_0) \tag{6.62}$$

where γ_i is a row vector with scalar coefficients. From (6.61) it follows that m satisfies

$$m[M_j] = \gamma_j N_j[:, M_j] = 0 \tag{6.63}$$

for all $j \in I$. This implies that $N_e(s_0) = 0$, but this contradicts the assumption that $N_i(s)$ was irreducible. Hence (6.60) must be true and the theorem follows. □

The proof of Theorem 6.5.

Proof. The theorem follows by recursive use of Lemma 6.4 and that all MO set in M are found by computing all PO sets of decreasing redundancy until the redundancy is one. □

6.4 Diagnosing all Behavioral Models

So far in this chapter, we have only been concerned to fulfill (6.1) for one single behavioral mode. In the next sections, we extend the discussion to the case when all behavioral modes in $\mathcal{B}$ are considered.

6.4.1 Sufficient to Test All MO Sets

In the next theorem it is stated that given a diagnosis model, it is sufficient to check the consistency of all feasible MO sets to determine the consistency of all behavioral models.

Theorem 6.6. *If $\mathbb{M}$ is a diagnosis model with a set $\boldsymbol{M} = \cup_{b \in \mathcal{B}} M_b$ of linear dynamic equations such that M_b is in the form* (6.2) *and $[H\,L]$ has full row rank for all $b \in \mathcal{B}$, then the set ω of all feasible MO sets satisfies* (6.1) *for all $b \in \mathcal{B}$.*

Proof. Let b be an arbitrary behavioral mode $b \in \mathcal{B}$. Any MO sets $M \subseteq M_b$ is a feasible MO set, i.e., $M \in \omega$. Therefore, Corollary 6.1 implies that (6.1) is satisfied for b. Since b was arbitrarily chosen the theorem follows. □

Before we give an example of a diagnosis model and all its MO sets we will first describe how all MO sets can be found in the next section.

6.5 Algorithm for Finding All MO Sets in a Linear Dynamic Model

An algorithm, i.e., Algorithm 7, for finding all feasible MO sets is a diagnosis model with linear static equations was given in Chapter 4. This algorithm consists of two main parts. One part, Algorithm 4, finds all MO sets in a linear static models (4.1) and the other part described in Chapter 5 handles equation assumptions such that only feasible MO sets are found. The part described in Chapter 5 is not dependent on the type of equations that the diagnosis model contains and can therefore be applied to any type of equations, also to linear differential equations.

Algorithm 4 for finding all MO sets of equations is a linear static models (4.1) can with small modifications also handle linear dynamic models of the form (6.2). The only difference is to interpret the matrices H and L as polynomial matrices. Next an example illustrates both how Algorithm 7 is applied to a dynamic linear model and how the output can be interpreted by considering the result of Theorem 6.6.

Example 6.9

Consider the electrical circuit in Example 3.17. Assume that resistor $R2$ is replaced with an inductor $L2$ with inductance L_2 and that the battery B is replaced with a power supply P that in the no fault mode generates a known time-varying voltage $V(t)$. Furthermore, assume that a faulty power supply implies that $V(t) = 0$, that is, similar to battery fault in Example 3.17. The model equations becomes

$$\begin{array}{llcr}
 & I - I_1 - I_2 & = 0 & (1) \\
R1 = NF & V - I_1 R_1 & = 0 & (2) \\
L2 = NF & V - L_2 \dot{I}_2 & = 0 & (3) \\
P = NF & V - U & = 0 & (4) \\
S1 = NF & V - y_V & = 0 & (5) \\
S2 = NF & I - y_I & = 0 & (6) \\
S3 = NF & I_2 - y_{I_2} & = 0 & (7) \\
P = F & V & = 0 & (8)
\end{array} \tag{6.64}$$

Note that the only difference between (3.95) and (6.64) is equation 3. The model (6.64) can be written using polynomial matrices as

$$\overbrace{\begin{bmatrix} 0 & 1 & -1 & -1 \\ 1 & 0 & -R_1 & 0 \\ 1 & 0 & 0 & -pL_2 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}^{H(p)} \begin{bmatrix} V \\ I \\ I_1 \\ I_2 \end{bmatrix} + \overbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \end{bmatrix}}^{L} \begin{bmatrix} U \\ y_V \\ y_I \\ y_{I_2} \end{bmatrix} = 0 \tag{6.65}$$

If Algorithm 7 is applied to this model, the following MO sets and their corresponding matrices $N_{H[M]}L$ are found to be:

$$\begin{array}{llrrrrl}
MO & & U & y_V & y_I & y_{I2} & \\
\hline
\{1,2,3,4,6\} & [& -(R_1 + pL_2) & 0 & pR_1L_2 & 0 &] \\
\{1,2,3,5,6\} & [& 0 & -(R_1 + pL_2) & pR_1L_2 & 0 &] \\
\{1,2,3,6,7\} & [& 0 & 0 & R_1 & -(R_1 + pL_2) &] \\
\{1,2,4,6,7\} & [& -1 & 0 & R_1 & -R_1 &] \\
\{1,2,5,6,7\} & [& 0 & -1 & R_1 & -R_1 &] \\
\{3,4,7\} & [& -1 & 0 & 0 & pL_2 &] \\
\{3,5,7\} & [& 0 & -1 & 0 & pL_2 &] \\
\{4,5\} & [& 1 & -1 & 0 & 0 &] \\
\{1,2,3,6,8\} & [& 0 & 0 & p & 0 &] \\
\{1,2,6,7,8\} & [& 0 & 0 & 1 & -1 &] \\
\{3,7,8\} & [& 0 & 0 & 0 & p &] \\
\{5,8\} & [& 0 & 1 & 0 & 0 &]
\end{array} \tag{6.66}$$

These are the same MO sets as obtained in Example 3.17. From Theorem 6.6, it follows these MO sets can be used to construct a sound and complete diagnosis system.

6.6 Minimizing The Number of Tests

In the linear static case, we have shown that it is not in general necessary to include all feasible MO sets in ω to satisfy (6.1) for all $b \in \mathcal{B}$. This holds true also for the linear dynamic case. For the static case, Algorithm 2 computes all minimal sets ω that solves (3.89). In this section we will modify this algorithm such that it can be applied to linear dynamic models.

The only part in Algorithm 2 that does not work also in the linear dynamic case is where Algorithm 1 is used. The objective of this algorithm is given a PO set B_i and the set $\mathcal{M}_{MO}$ of all MO models, compute the set Ω'_i of all minimal solutions $\omega' \subseteq \mathcal{M}_{MO}$ of (3.90).

These computations are based on the result given in Theorem 3.6, i.e., given a PO set B_i and a family of subsets $\omega' = \{M_1, M_2, \dots M_n\}$ where $M_j \subseteq B_i$ for all $M_j \in \omega'$, it holds that

$$\mathcal{O}(B_i) = \bigcap_{M \in \omega'} \mathcal{O}(M) \tag{6.67}$$

if and only if

$$\varphi(\omega') = \varphi(B_i) \tag{6.68}$$

Let Definition 3.5 of $\varphi(\omega')$ be extended to polynomial matrices. For the dynamic case, it then holds that (6.67) implies (6.68) but the converse implication is not true according to Theorem 6.4. To get equivalence with (6.67), the root criteria (6.42) must in addition to (6.68) be satisfied, i.e., that the matrix

$$\begin{bmatrix} N^T_{H[M_1]} & N^T_{H[M_2]} & \cdots & N^T_{H[M_n]} \end{bmatrix}$$

has no roots.

A consequence of this is that it is not sure that a minimal solution of (3.90) contains exactly $\varphi(B_i)$ MO sets. In the next algorithm also larger sets may be consider if the minimal sets are not found among those with size $\varphi(B_i)$. The next algorithm is used in the dynamic case instead of Algorithm 1 in Algorithm 2 to obtain all minimal solutions $\omega' \subseteq \mathcal{M}_{MO}$ of (3.90).

Algorithm 8. Ω'_i = `MinimalTestSets`$(B_i, \mathcal{M}_{MO})$

```
$\Omega'_i = \emptyset$;
$\alpha := \{M \in \mathcal{M}_{MO} | M \subseteq B_i\}$;
size := $\varphi(B_i)$; $\hat{\Omega} := \{\omega' \subseteq \alpha \,|\, |\omega'| = $ size$\}$;
while $\hat{\Omega} \neq \emptyset$ do
    for each $\omega' \in \hat{\Omega}$ do
        if $\varphi(\omega') = \varphi(B_i)$ and (6.42) is true do
            Insert $\omega'$ in $\Omega'_i$;
        end if
    end for
    size := size + 1;
    $\hat{\Omega} := \{\omega' \subseteq \alpha \,|\, |\omega'| = $ size, $\omega'$ is no superset of any set in $\Omega'_i\}$;
end while
return $\Omega'_i$
```

Given a PO set B_i and the set $\mathcal{M}_{MO}$ of all MO models, the output set Ω'_i in Algorithm 8 contains all minimal solutions $\omega' \subseteq \mathcal{M}_{MO}$ of (3.90).

Next, we will discuss the result of some examples.

Example 6.10
Continuation of Example 6.9. If all multiple fault modes are considered, there is only one minimal solution and this solution is equal to the MO sets obtained in Example 3.17.

6.7 Fault Influence on Residuals

In this section we extend the main results concerning fault influence on residuals given in Section 3.10 to the linear dynamic case.

Theorem 6.7. *Let M be an MO set of linear differential equations*

$$H(p)\,x + L(p)\,z = 0 \tag{6.69}$$

and let model deviation be described by a vector ϵ such that

$$H(p)\,x + L(p)\,z = \epsilon \tag{6.70}$$

For a residual

$$r = N_H(p)L(p)z \tag{6.71}$$

the residual response of the model deviation is given by

$$r = N_H(p)\epsilon \tag{6.72}$$

where $N_H[:, \{i\}] \neq 0$ for all $i \in M$.

Proof. Since (6.69) is an MO sets, it has redundancy one. This means that $N_H(p)$ is a vector determined up to a non-zero constant. By multiplying (6.70) with $N_H(p)$ from the left, we get both the residual computational form (6.71) and the fault response (6.72). The results of Corollary 3.1 can be proven analogously for the dynamic case with polynomial matrices. Since M is an MO set Corollary 3.1 states that all columns in $N_H(p)$ are non-zero, i.e., $N_H[:, \{i\}] \neq 0$ for all $i \in M$. □

6.7.1 Fault Influence and Null Hypothesis

In the static case, a residual r derived from M is said to be influenced by a behavior mode b if $r \neq 0$ is consistent with the behavioral model M_b. This definition is applicable also to the dynamic case.

The next theorem gives a sufficient condition on the diagnosis model $\mathbb{M}$ such that the fault influence of any residual based on an MO set M is given by the equation assumptions, i.e., the fault influence is $(\text{assump}(M))^C$.

Theorem 6.8. *Let $\mathbb{M}$ be a diagnosis model with a set $\boldsymbol{M}$ of equations*

$$H(p)x + L(p)z = 0 \tag{6.73}$$

Let the model deviation be modeled as

$$H(p)x + L(p)z = F\epsilon \tag{6.74}$$

where F is a $|\boldsymbol{M}| \times |\boldsymbol{M}|$ matrix defined by

$$F_{ij} = \begin{cases} 1 & \textit{if } i = j \textit{ and } \mathrm{assump}(e_i) \neq \mathcal{B} \\ 0 & \textit{otherwise} \end{cases} \tag{6.75}$$

If

$$\mathrm{Im}(F) \subseteq \mathrm{Im}([H\,L]) \tag{6.76}$$

then for any MO set $M \subseteq \boldsymbol{M}$ of equations and for any residual

$$r = N_{H[M]}L(p)z \tag{6.77}$$

not identically zero, r is influenced by all behavioral modes in $(\mathrm{assump}(M))^C$ *and no others.*

Proof. An MO set M and a residual r are given. We start to prove that r is not influenced by any behavioral mode $b \in \mathrm{assump}(M)$. By the definition of the operator assump, it follows that

$$\mathrm{sys} \in \mathrm{assump}(M) \rightarrow z \in \mathcal{O}(M) \tag{6.78}$$

The observation set $\mathcal{O}(M)$ can according to (6.15) and (6.77) be expressed as

$$\mathcal{O}(M) = \{z | N_{H[M]}L(p)z = 0\} = \{z | r(z) = 0\} \tag{6.79}$$

The formulas (6.78) and (6.79) imply that

$$\mathrm{sys} \in \mathrm{assump}(M) \rightarrow r = 0 \tag{6.80}$$

i.e., r is not influenced by any behavioral mode $b \in \mathrm{assump}(M)$.

Now, we continue to prove that r is influenced by all behavioral modes $b \notin \mathrm{assump}(M)$. The idea is to take an arbitrary behavioral mode $b \in (\mathrm{assump}(M))^C$ and show that the equations M_b and $r \neq 0$ are consistent. From the definition of the operator assump, it follows for any set $\bar{M}$ that

$$\bar{M} \subseteq M_b \rightarrow b \in \mathrm{assump}(\bar{M})$$

Then, since $b \notin \mathrm{assump}(M)$, it follows that $M \nsubseteq M_b$ or equivalently that $M \setminus M_b \neq \emptyset$. Let $M \setminus M_b$ be denoted by M'. By construction of M', it follows that $b \notin \mathrm{assump}(e)$ for any $e \in M'$. This implies that $\mathrm{assump}(e) \neq \mathbf{B}$ for all $e \in M'$, i.e., $F[M', M']$ is the identity matrix.

From (6.74), (6.77), and Theorem 6.7, we get the fault influence

$$r = N_{H[M]}F\epsilon \tag{6.81}$$

The only equations $e \in M$ that might be inconsistent in b, i.e., $F[\{e\}]\epsilon \neq 0$, are the equations $e \in M'$. By the construction of F in (6.75) and the fact that $\mathrm{assump}(e) \neq \mathbf{B}$, it follow that

$$F[\{e\}]\epsilon = \epsilon[\{e\}] \tag{6.82}$$

for all $e \in M'$. Hence, (6.81) can be rewritten as

$$r = N_{H[M]}F\epsilon = N_{H[M]}[:, M']F[M', M']\epsilon[M'] = N_{H[M]}[:, M']\epsilon[M'] \tag{6.83}$$

From (6.76), it follows that

$$\mathrm{Im}(F[M \cup M_b]) \subseteq \mathrm{Im}([H[M \cup M_b]\, L[M \cup M_b]]) \tag{6.84}$$

i.e., for any $\boldsymbol{\epsilon}[M \cup M_b]$ there exist some $\boldsymbol{x}$ and $\boldsymbol{z}$ such that the subset $M \cup M_b$ of the equations in (6.74) is satisfied. Since $M' \cap M_b = \emptyset$, we can choose $\boldsymbol{\epsilon}[M_b] = 0$, i.e.,

$$N_{H[M_b]} L\boldsymbol{z} = 0 \tag{6.85}$$

and $\boldsymbol{\epsilon}[M']$ such that

$$r = N_{H[M]} L\boldsymbol{z} \neq 0 \tag{6.86}$$

in (6.83). The expressions (6.86) and (6.85), imply that the equations M_b and $r \neq 0$ are consistent, i.e., r is influenced by b. Since b was an arbitrarily chosen behavioral modes such that $b \notin \mathrm{assump}(M)$, it follows that r is influenced by all behavioral modes in $(\mathrm{assump}(M))^C$ and the theorem follows. □

Note that a sufficient condition for (6.76) is that $[H\, L]$ has full row-rank. This is often true for models with one maximal feasible model and then Theorem 6.8 is applicable.

Example 6.11
Consider the electrical circuit in Example 6.9. For the model (6.65), it holds that $[H\, L]$ has full row rank. This is a sufficient condition for (6.76) and it follows that for any residual r based on an MO set M, $(\mathrm{assump}(M))^C$ are the behavioral modes that r is influenced by.

6.8 Conclusions

In Chapter 2 we showed that one strategy to construct a diagnosis system was to start with a diagnosis model $\mathbb{M}$ and choose a set $\omega = \{M_1, \ldots, M_n\}$ of rejectable models to test. There, it was also shown that a diagnosis system based on ω can be sound if and only if the set ω fulfills (6.1) for all behavioral modes $b \in \mathbf{B}$.

This chapter has presented theory and algorithms for linear dynamic models corresponding to the presentation in Chapter 3 for linear static models. A key result is that if ω is chosen to be the set of all feasible MO sets in the diagnosis model $\mathbb{M}$, then ω fulfills (6.1) for all behavioral modes $b \in \mathbf{B}$ according to Theorem 6.6. All these MO sets can be found by using Algorithm 7. It has also been shown that it is not in general necessary to include all MO sets in ω to satisfy (6.1) for all behavioral modes $b \in \mathbf{B}$. Theory for selecting MO sets has been developed and a key result for this is the conditions given in Theorem 6.4. In addition to the rank-condition valid for the static case, the root condition (6.42) must also be added. The root condition is then used to modify Algorithm 2 such that it is applicable also to dynamic models. The output of the modified algorithm contains all minimal subsets ω that fulfill (6.1) for all behavioral modes $b \in \mathbf{B}$. A minimal cardinality set of MO sets can then be picked out from the set of all minimal sets and this set corresponds to a sound and complete diagnosis system with the minimum number tests.

Finally, Theorem 6.8 showed that under a mild rank condition on the diagnosis model and given an MO set M, the behavioral modes that influence any residual derived from M are given by the equation assumptions according to (assump(M))^C. Hence if the rank condition of the model can be verified, then any further fault influence analysis of each residual is not needed.

Appendix

Polynomial matrices have been shown to be a useful representation of linear differential equations. In this way the constant matrices representing static models can be extended to polynomial matrices for differential equations. Properties of polynomial matrices representing dynamic systems have been extensively studied, e.g. in (Kailath, 1980; Polderman and Willems, 1998). This Appendix recalls some basic definitions and properties concerning polynomial matrices. All these can be found in the references mentioned above.

6.A Some Properties of Polynomial Matrices

A *polynomial matrix* is a matrix where each individual element is a scalar polynomial in s with coefficients in any field, but in this thesis we will always have real coefficients. An example of a polynomial matrix is

$$M(s) = \begin{bmatrix} 1 & -(s-1) & 0 \\ s & 0 & -(s-1) \\ 0 & s & -1 \end{bmatrix} \tag{6.87}$$

Definition 6.A.5 (Normal Rank). *The* ***normal rank*** *of a polynomial matrix $P(s)$ is the maximal rank that $P(s)$ has for any $s \in \mathbb{C}$.*

The matrix $M[\{1,2\}]$ in (6.87) has normal rank two, i.e., it has full row rank in the normal rank sense. When there is no risk for confusion we will drop the word normal in front of rank and use only rank. The matrix $M(s)$ has rank two and the rows are linearly dependent, because there exists a linear combination with polynomial coefficients that sums up to zero, i.e.,

$$\begin{bmatrix} s & -1 & s-1 \end{bmatrix} M(s) = 0 \tag{6.88}$$

We say that the rows of $M(s)$ are *linearly dependent*.

Definition 6.A.6 (Roots). *The* ***roots*** *of a polynomial matrix $M(s) \in \mathbb{R}^{m\times n}[s]$ are those points in the complex plane $s \in \mathbb{C}$ where $M(s)$ loses rank.*

If $M(s)$ is square then its roots are the roots of its determinant $\det M(s)$, including multiplicity.

Definition 6.A.7 (Irreducible Polynomial Matrix). *A polynomial matrix $M(s) \in \mathbb{R}^{m\times n}[s]$ is an* ***irreducible polynomial matrix*** *if it has full rank, i.e.,* $\max(m,n)$*, for all (finite) values of $s \in \mathbb{C}$.*

From Definition 6.A.6 and Definition 6.A.7, it follows that a matrix is irreducible if and only if it has full normal rank and no roots. The matrix

$$U(s) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -(s-1) & 0 \\ 0 & -s & 1 \end{bmatrix} \tag{6.89}$$

is irreducible, because it has full row rank for all finite values of s.

Definition 6.A.8 (Unimodular Polynomial Matrix). *A square irreducible polynomial matrix is a* ***unimodular polynomial matrix****.*

The matrix $U(s)$ in (6.89) is unimodular, because it is square and irreducible. A matrix is unimodular if and only if its determinant is independent of s. For example, the determinant of $U(s)$ in (6.89) is $\det U(s) \equiv -1$.

Theorem 6.A.1. *The inverse of a polynomial matrix is unimodular if and only if its inverse is also polynomial.*

The inverse of a unimodular matrix is also unimodular. There are three *elementary row operations* for polynomial matrices and they are:

1. Interchange two rows.
2. Add to a row, a polynomial multiple of any other row.
3. Scale a row by a non-zero number in the coefficient field.

Each elementary row operation can be written as a left multiplication of a matrix. Three examples of matrices corresponding to each type of the elementary row operations are

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ -(s-1) & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.90}$$

Such matrices are called *elementary matrices* and these are unimodular matrices. It holds that a matrix is unimodular if and only if it is a product of elementary matrices.

6.A.1 Polynomial Basis

Definition 6.A.9 (Polynomial Basis). *A* ***polynomial basis*** *for a rational vector space $\mathcal{N}$ is a set of linearly independent vectors in $\mathcal{N}$ such that these vectors span the space $\mathcal{N}$.*

A polynomial basis can be represented by a matrix $M(s)$ where each row corresponds to a vector in the basis. The highest degree of all polynomials in a vector is called the *degree of the vector* and *row degree* if the vector is a row in a polynomial matrix. The degrees of the polynomials in the first row of $M(s)$ in (6.87) are zero, one, and $-\infty$. This implies that the row degree is one. In fact, all row degrees of $M(s)$ are one. The *order* of a matrix is defined as the sum of the row-degrees of all its rows.

Definition 6.A.10 (Minimal Polynomial Basis). *A* ***minimal polynomial basis*** *for $\mathcal{N}$ is a polynomial basis for $\mathcal{N}$ with minimum order.*

Theorem 6.A.2. *A minimal polynomial basis is an irreducible basis.*

Theorem 6.A.3. *If the rows of $N(s)$ form an irreducible polynomial basis for a vector space $\mathcal{N}$, then any polynomial row vector $n(s) \in \mathcal{N}$ can be written $n(s) = \phi(s)N(s)$ where $\phi(s)$ is a polynomial row vector.*

7

S D G S

Following the strategy described in Section 2.3.3, the construction of a diagnosis system starts with finding a set $\omega = \{M_1, \ldots, M_n\}$ of rejectable models to test. If the diagnosis system should be sound, the set ω must fulfill

$$\mathcal{O}(M_b) = \bigcap_{M_i \in \omega : M_i \subseteq M_b} \mathcal{O}(M_i) \tag{7.1}$$

for all $b \in \mathbf{B}$ according to Theorem 2.2. In Chapter 6 it was shown in Theorem 6.6 that it is sufficient to include all MO sets in ω to fulfill (7.1) for all $b \in \mathbf{B}$.

In this chapter, the model equations are assumed to be non-linear static or dynamic equations. A result similar to the result in Theorem 6.6 given for linear models will here be developed for non-linear models. This is a significantly more difficult problem and it will be assumed that rejectable models corresponding to MO sets can be computed. Given these models, a key result is a necessary and sufficient condition for which sets ω of models that must be used to design a sound and complete diagnosis system. Using this result, it is possible to calculate a set of minimum number of models that corresponds to a sound and complete diagnosis system.

First, model definitions are presented in Section 7.1 before two different solutions to (7.1) are given in Section 7.2 and Section 7.3 respectively. Then, a necessary and sufficient condition for a set ω to fulfill (7.1) is given in Section 7.4. Finally the conclusions are drawn.

7.1 Model Definitions

Two model properties applicable to non-linear models have been defined earlier in this thesis, that is feasible model defined in Definition 2.4 and rejectable model defined in Definition 2.5. To state a result corresponding to Theorem 6.6 for non-linear models, a non-linear version of the model property MO set needs to be defined. The last of the following three model definitions will be the non-linear correspondence to MO set.

Definition 7.1 (Rejectable Model at z_0). *A set M of equations is a **rejectable model at** z_0 if*

$$z_0 \notin \mathcal{O}(M) \tag{7.2}$$

Definition 7.2 (Minimal Rejectable Model at z_0). *A set M of equations is a **minimal rejectable model at** z_0 if no proper subset of M is a rejectable model at z_0.*

Definition 7.3 (Minimal Rejectable Model). *A set M of equations is a **minimal rejectable model** if there exists a z such that M is a minimal rejectable model at z.*

The set of all feasible minimal rejectable models in $\mathbb{M}$ at z_0 is denoted $\omega_m(z_0)$ and the set of all feasible minimal rejectable models in $\mathbb{M}$ is denoted ω_m.

For linear models, a model is an MO set if and only if it is a minimal rejectable model. Furthermore, a linear model is a minimal rejectable model if and only if the model is a minimal rejectable model at some z. However, the latter is not true for a general non-linear model and this will be shown by the next example.

Example 7.1

Consider the following diagnosis model:

Assumption	Equation	Expression
Sensor 1		
$s_1 = \text{NF}$	e_1	$z_1 = x_1$
Comp		
	e_2	$x_1 = x_2^2$
Sensor 2		
$s_2 = \text{NF}$	e_3	$z_2 = x_2$
Sensor 3		
$s_3 = \text{NF}$	e_4	$z_3 = x_2$
$s_3 = \text{SG}$	e_5	$z_3 = 0$

(7.3)

with the possible component behavioral modes defined by:

Component	Possible behavioral modes
Sensor 1	$s_1 \in \{\text{NF}, \text{UF}\}$
Sensor 2	$s_2 \in \{\text{NF}, \text{UF}\}$
Sensor 3	$s_3 \in \{\text{NF}, \text{SG}\}$

(7.4)

Let $z_0 = (z_1, z_2, z_3)$ be such that $z_1 < 0$ and $z_2 = z_3 \neq 0$. The (feasible) rejectable models at z_0 are

$$\begin{gathered}\{e_5\}, \{e_1, e_2\}, \{e_1, e_5\}, \{e_2, e_5\}, \{e_3, e_5\}, \\ \{e_1, e_2, e_3\}, \{e_1, e_2, e_4\}, \{e_1, e_2, e_5\}, \{e_1, e_3, e_5\}, \\ \{e_2, e_3, e_5\}, \{e_1, e_2, e_3, e_4\}, \{e_1, e_2, e_3, e_5\}\end{gathered} \tag{7.5}$$

An example of a model that is not a rejectable model at z_0 is $\{e_3, e_4\}$. The minimal rejectable models at z_0 are

$$\omega_m(z_0) = \{\{e_1, e_2\}, \{e_5\}\} \tag{7.6}$$

An example of a rejectable model at z_0 that is not a minimal rejectable model at z_0 is $\{e_1, e_5\}$ because the proper subset $\{e_5\}$ is a rejectable model at z_0.

The minimal rejectable models are

$$\omega_m = \{\{e_5\}, \{e_3, e_4\}, \{e_1, e_2\}, \{e_1, e_2, e_3\}, \{e_1, e_2, e_4\}\} \tag{7.7}$$

Note that both $\{e_1, e_2\}$ and $\{e_1, e_2, e_3\}$ are minimal rejectable models even though $\{e_1, e_2\} \subset \{e_1, e_2, e_3\}$. To explain this, let $z_1 = (z_1, z_2, z_3)$ be such that $z_1 \geq 0$, $z_1 \neq z_2^2$, and $z_2 = z_3 \neq 0$. The minimal rejectable models at z_1 are

$$\omega_m(z_1) = \{\{e_1, e_2, e_3\}, \{e_1, e_2, e_4\}, \{e_5\}\} \tag{7.8}$$

The model $\{e_1, e_2, e_3\}$ is a minimal rejectable model at z_1 and one of its subset $\{e_1, e_2\}$ is a minimal rejectable model at z_0. According to Definition 7.3 it holds that both $\{e_1, e_2\}$ and $\{e_1, e_2, e_3\}$ are minimal rejectable models. Later, we will show that all minimal rejectable models of are needed to obtain soundness. An example of a model that is not a minimal rejectable model is $\{e_1, e_2, e_3, e_4\}$. This model is for example rejectable at z_0 as seen in (7.5) but if it is rejectable at an arbitrary z, then there is always a proper subset that is rejectable at z too.

In the following sections we will describe different sets ω that satisfies (7.1).

7.2 All Behavioral Models Sufficient for Soundness

In this section, we show the elementary result that a sound diagnosis system can be based on the set of all behavioral models in a diagnosis model.

Theorem 7.1. *If* $\mathbb{M}$ *is a diagnosis model with a set* $\boldsymbol{M} = \cup_{b \in \mathbf{B}} M_b$ *of non-linear equations, then the set* ω *of all system behavioral models* M_b *satisfies* (7.1) *for all* $b \in \mathbf{B}$.

Proof. Follows trivially from Theorem 2.2. □

Next an example shows how the design of a sound and complete diagnosis system, based on the result in Theorem 7.1, can be done.

Example 7.2
The behavioral models in (7.3) are

b	M_b
$\langle NF, NF, NF \rangle$	$\{e_1, e_2, e_3, e_4\}$
$\langle UF, NF, NF \rangle$	$\{e_2, e_3, e_4\}$
$\langle NF, UF, NF \rangle$	$\{e_1, e_2, e_4\}$
$\langle NF, NF, SC \rangle$	$\{e_1, e_2, e_3, e_5\}$
$\langle UF, UF, NF \rangle$	$\{e_2, e_4\}$
$\langle UF, NF, SC \rangle$	$\{e_2, e_3, e_5\}$
$\langle NF, UF, SC \rangle$	$\{e_1, e_2, e_5\}$
$\langle UF, UF, SC \rangle$	$\{e_2, e_5\}$

(7.9)

According to Theorem 7.1, a complete and sound diagnosis system for (7.3) can be derived and an example is

Δ	M_i	$H_i^0 : sys \in \Phi_i = \text{assump}(M_i)$	$\{z \mid T_i(z) \notin \mathcal{R}_i\} = \mathcal{O}(M_i)$
δ_1	$\{e_1, e_2, e_3, e_4\}$	$\phi(s_1 = NF \land s_2 = NF \land s_3 = NF)$	$\{z \mid z_1 = z_2^2, z_2 = z_3\}$
δ_2	$\{e_2, e_3, e_4\}$	$\phi(s_2 = NF \land s_3 = NF)$	$\{z \mid z_2 = z_3\}$
δ_3	$\{e_1, e_2, e_4\}$	$\phi(s_1 = NF \land s_3 = NF)$	$\{z \mid z_1 = z_3^2\}$
δ_4	$\{e_1, e_2, e_3, e_5\}$	$\phi(s_1 = NF \land s_2 = NF \land s_3 = SC)$	$\{z \mid z_1 = z_2^2, z_3 = 0\}$
δ_5	$\{e_2, e_4\}$	$\phi(s_2 = NF)$	$\mathbb{R}^3$
δ_6	$\{e_2, e_3, e_5\}$	$\phi(s_2 = NF \land s_3 = SC)$	$\{z \mid z_3 = 0\}$
δ_7	$\{e_1, e_2, e_5\}$	$\phi(s_1 = NF \land s_3 = SC)$	$\{z \mid z_1 \geq 0, z_3 = 0\}$
δ_8	$\{e_2, e_5\}$	$\phi(s_3 = SG)$	$\{z \mid z_3 = 0\}$

(7.10)

Note that the null hypothesis of test δ_5 is not rejectable since $\mathcal{O}(M_5) = \mathbb{R}^3$. Therefore it is possible to omit δ_5. If exactly the diagnosis system (7.10) was implemented we would for example check if $z_3 = 0$ in 4 out of the 8 tests. This is computationally not an efficient way to diagnose the system and we will see in the next section that a better alternative is to use minimal rejectable models.

7.3 All Minimal Rejectable Models are Sufficient for Soundness

In this section, we show that a sound diagnosis system can be based on the set of all minimal rejectable models in a diagnosis model. As the name indicates they are the smallest models that can be used to obtain soundness and testing small models have advantages according to the discussion in Section 2.6.

Theorem 7.2. *Given a model M, let ω be the set of all minimal rejectable models in M. Then it follows that*

$$\mathcal{O}(M) = \bigcap_{E \in \omega} \mathcal{O}(E) \tag{7.11}$$

Proof. The equality (7.11) holds, if both

$$\mathcal{O}(M) \subseteq \bigcap_{E \in \omega} \mathcal{O}(E) \tag{7.12}$$

and

$$\bigcap_{E \in \omega} \mathcal{O}(E) \subseteq \mathcal{O}(M) \tag{7.13}$$

hold. Inclusion (7.12) is always true because for any set $E \in \omega$, $E \subseteq M$ implies that $\mathcal{O}(M) \subseteq \mathcal{O}(E)$. Hence it remains to prove that (7.13) holds.

We will prove (7.13) by showing the equivalent statement that for all z such that

$$z \notin \mathcal{O}(M) \tag{7.14}$$

it follows that

$$z \notin \bigcap_{E \in \omega} \mathcal{O}(E) \tag{7.15}$$

Let z_0 be an arbitrary z that fulfills (7.14). From the definition of minimal rejectable model at z_0, there exists a minimal rejectable model M_1 at z_0 such that

$$M_1 \subseteq M \tag{7.16}$$

and

$$z_0 \notin \mathcal{O}(M_1) \tag{7.17}$$

Note that the existence follows from the fact that if no proper subset to M has property (7.17), then M is a minimal rejectable model at z_0. Since all minimal rejectable models at z_0 are minimal rejectable models, it follows that $M_1 \in \omega$. This implies that

$$\bigcap_{E \in \omega} \mathcal{O}(E) \subseteq \mathcal{O}(M_1) \tag{7.18}$$

From (7.17) and (7.18), we get (7.15) with $z = z_0$. Since z_0 was arbitrarily chosen, it follows that (7.14) implies (7.15) and equivalently that (7.13) holds. This completes the proof. □

Corollary 7.1. *If* $\mathbb{M}$ *is a diagnosis model with a set* $\mathbf{M} = \cup_{b \in \mathbf{B}} M_b$ *of equations, then the set* $\omega = \omega_m$ *of all feasible minimal rejectable models in* $\mathbf{M}$ *satisfies* (7.1) *for all* $b \in \mathbf{B}$.

Proof. Let b_0 be an arbitrary behavioral mode in $\mathbf{B}$. For this behavioral mode, the condition (7.1) becomes

$$\mathcal{O}(M_{b_0}) = \bigcap_{M_i \in \omega : M_i \subseteq M_{b_0}} \mathcal{O}(M_i) \tag{7.19}$$

The sets $M_i \in \omega$ such that $M_i \subseteq M_{b_0}$ specified by this intersection are the minimal rejectable models included in M_{b_0}. Then it follows that (7.19) is true according to Theorem 7.2. Since $b_0 \in \mathbf{B}$ was arbitrarily chosen, the result follows. □

Example 7.3

The minimal rejectable models in (7.3) are

$$\omega_m = \{\{e_1, e_2\}, \{e_1, e_2, e_3\}, \{e_1, e_2, e_4\}, \{e_3, e_4\}, \{e_5\}\} \tag{7.20}$$

A complete and sound diagnosis system for (7.3) is

Δ	$H_i^0 : \Phi_i = \text{assump}(M_i)$	M_i	$\{z \mid T_i(z) \notin \mathcal{R}_i\} = \mathcal{O}(M_i)$
δ_1	$\phi(s_1 = NF)$	$\{e_1, e_2\}$	$\{z \mid z_1 \geq 0\}$
δ_2	$\phi(s_1 = NF \wedge s_2 = NF)$	$\{e_1, e_2, e_3\}$	$\{z \mid z_1 = z_2^2\}$
δ_3	$\phi(s_1 = NF \wedge s_3 = NF)$	$\{e_1, e_2, e_4\}$	$\{z \mid z_1 = z_3^2\}$
δ_4	$\phi(s_2 = NF \wedge s_3 = NF)$	$\{e_3, e_4\}$	$\{z \mid z_2 = z_3\}$
δ_5	$\phi(s_3 = SG)$	$\{e_5\}$	$\{z \mid z_3 = 0\}$

(7.21)

A comparison between (7.10) and (7.21) reveals that the number of tests in (7.21) is smaller, the tests in (7.21) contains less number of equations, and the sets $\mathcal{O}(M_i)$ in (7.21) are described with simpler expressions.

Table 7.1: All minimal sets σ_b for each system behavioral mode b in the diagnosis model defined in (7.3).

b	M_b	Σ_b
b_1	$\{e_1, e_2, e_3, e_4\}$	$\{\{\{e_1, e_2, e_3\}, \{e_3, e_4\}\}, \{\{e_1, e_2, e_4\}, \{e_3, e_4\}\}\}$
b_2	$\{e_2, e_3, e_4\}$	$\{\{\{e_3, e_4\}\}\}$
b_3	$\{e_1, e_2, e_4\}$	$\{\{\{e_1, e_2, e_4\}\}\}$
b_4	$\{e_1, e_2, e_3, e_5\}$	$\{\{\{e_1, e_2, e_3\}, \{e_5\}\}\}$
b_5	$\{e_2, e_4\}$	$\{\emptyset\}$
b_6	$\{e_2, e_3, e_5\}$	$\{\{\{e_5\}\}\}$
b_7	$\{e_1, e_2, e_5\}$	$\{\{\{e_1, e_2\}, \{e_5\}\}\}$
b_8	$\{e_2, e_5\}$	$\{\{\{e_5\}\}\}$

(7.23)

7.4 A Sufficient and Necessary Condition for Soundness

Finally, a sufficient and necessary condition of the set of models that can be used to derive a sound diagnosis system is given.

Let the set of all minimal rejectable models in a behavioral model M_b be denoted by ω_b. In general, not all minimal rejectable models in the set M_b is needed to satisfy (7.1). Therefore, let $\sigma_b \subseteq \omega_b$ be a minimal set such that

$$\mathcal{O}(M_b) = \bigcap_{M \in \sigma_b} \mathcal{O}(M) \tag{7.22}$$

Let the sets of all such sets be denoted by Σ_b. The sets σ_b and Σ_b are exemplified next.

Example 7.4
Consider the diagnosis model in (7.3). The minimal rejectable models in this model are given in (7.7). The minimal rejectable models in each system behavioral mode are given by $\omega_b = \{M \in \omega_m | M \subseteq M_b\}$. The different sets Σ_b are shown in Table 7.1. In this example $\sigma_b = \omega_b$ for all behavioral modes except for b_1, b_3, and b_4.

Now, we are ready to give a characterization of sets ω that fulfills (7.1) for all $b \in \mathbf{B}$.

Theorem 7.3 (Sound Diagnosis System). *Let $\mathbb{M}$ be a diagnosis model with a set $\boldsymbol{M} = \cup_{b \in \mathbf{B}} M_b$ of equations, and let Σ_b be defined as above. Then a set ω fulfills* (7.1) *for all $b \in \mathbf{B}$ if and only if ω fulfills*

$$\exists \sigma_b \in \Sigma_b \forall M' \in \sigma_b \exists M \in \omega : M' \subseteq M \subseteq M_b \tag{7.24}$$

for all $b \in \mathbf{B}$.

Next Theorem 7.3 is proved and then two examples follow.

Proof. We start to show the if-direction, that is, if ω fulfills (7.24) for a behavioral mode $b_0 \in \mathbf{B}$, then ω fulfills also (7.1) for b_0. The equality (7.1) holds, if both

$$\mathcal{O}(M_{b_0}) \subseteq \bigcap_{M_i \in \omega : M_i \subseteq M_{b_0}} \mathcal{O}(M_i) \tag{7.25}$$

and

$$\bigcap_{M_i \in \omega : M_i \subseteq M_{b_0}} O(M_i) \subseteq O(M_{b_0}) \tag{7.26}$$

hold. Inclusion (7.25) is always true because for all $M_i \in \omega$ such that $M_i \subseteq M_{b_0}$ it holds that $O(M_{b_0}) \subseteq O(M_i)$. Hence it remains to prove that (7.26) holds for b_0 if (7.24) is satisfied for b_0.

Let z_0 be an arbitrary z such that

$$z_0 \in \bigcap_{M_i \in \omega : M_i \subseteq M_{b_0}} O(M_i) \tag{7.27}$$

From this we get that for all $M_i \in \omega$ such that $M_i \subseteq M_{b_0}$, it holds that

$$z_0 \in O(M_i) \tag{7.28}$$

From (7.24) it follows that there is a $\hat{\sigma}_{b_0} \in \Sigma_{b_0}$ such that

$$\forall M' \in \hat{\sigma}_{b_0} \exists M \in \omega : M' \subseteq M \subseteq M_{b_0} \tag{7.29}$$

holds. Expression (7.28), and (7.29) imply that

$$z_0 \in O(M') \tag{7.30}$$

for all $M' \in \hat{\sigma}_{b_0}$ or equivalently that

$$z_0 \in \bigcap_{M' \in \hat{\sigma}_{b_0}} O(M') \tag{7.31}$$

This and (7.22) imply that

$$z_0 \in O(M_{b_0}) \tag{7.32}$$

Hence, (7.26) is proved. Since z_0 and b_0 were arbitrarily chosen the if-direction follows.

Now, the only-if direction remains to be proven. Assume that (7.1) holds and let b_0 be an arbitrary behavioral mode in **B**. Let $\gamma_{b_0} = \{M \in \omega | M \subseteq M_{b_0}\}$ and enumerate the sets such that $\gamma_{b_0} = \{M_1, M_2, \ldots, M_i, \ldots, M_n\}$. Then it follows from (7.1) that

$$O(M_{b_0}) = \bigcap_{i=1}^{n} O(M_i) \tag{7.33}$$

If γ_i is the set of all minimal rejectable models in M_i, then Theorem 7.2 implies that

$$O(M_i) = \bigcap_{M' \in \gamma_i} O(M') \tag{7.34}$$

If (7.33) and (7.34) are combined, we get

$$O(M_{b_0}) = \bigcap_{M' \in \cup_{i=1}^{n} \gamma_i} O(M') \tag{7.35}$$

Now, the set $\hat{\sigma}_{b_0}$ will be chosen as an arbitrary minimal subset of $\cup_{i=1}^{n} \gamma_i$ such that

$$O(M_{b_0}) = \bigcap_{M' \in \hat{\sigma}_{b_0}} O(M') \tag{7.36}$$

Since each γ_i contains only minimal rejectable subsets of M_{b_0}, it follows that $\cup_{i=1}^{n}\gamma_i \subseteq \omega_{b_0}$. A minimal set that satisfies (7.36) and is a subset of $\cup_{i=1}^{n}\gamma_i$ is therefore also a minimal set that satisfies (7.36) and is a subset of ω_{b_0}. This and that $\hat{\sigma}_{b_0} \subseteq \cup_{i=1}^{n}\gamma_i$ is a minimal set that satisfies (7.36) imply that $\hat{\sigma}_{b_0} \in \Sigma_{b_0}$. Furthermore, by construction of $\hat{\sigma}_{b_0}$, it follows that for each $M' \in \hat{\sigma}_{b_0}$ there exists a set $M \in \omega$ such that $M' \subseteq M$. Hence (7.24) holds for b_0. Since b_0 was arbitrary chosen, the theorem follows. □

Example 7.5

Continuation of Example 7.4. According to Theorem 7.3 a sound diagnosis system can be obtained if and only if (7.24) holds for all $b \in \mathbf{B}$. Two particular sets ω that we studied earlier were $\omega_B = \{M_b | b \in \mathbf{B}\}$ and ω_m. Both these two sets satisfy (7.24) for all $b \in \mathbf{B}$ trivially.

Theorem 7.3 can be used to find the minimal number of tests that have to be used to design a sound and complete diagnosis system. The minimal number of tests for the diagnosis model described in (7.3) is 5. This can be realized from the following discussion. Row b_3 in Table 7.1 and condition (7.24) imply that there must be a set $M \in \omega$ such that $\{e_1, e_2, e_4\} \subseteq M \subseteq \{e_1, e_2, e_4\}$. Row b_8 implies that either $\{e_5\}$ or $\{e_2, e_5\}$ must be included in ω. Since $\{e_2, e_5\} \notin \sigma_b$ for any $b \in \mathbf{B}$, $\{b_5\}$ can be chosen. With $\omega = \{\{e_5\}, \{e_1, e_2, e_4\}\}$ condition (7.24) of b_3, b_5, b_6, and b_8 are fulfilled. Continuing in this way the minimum number of 5 models must be included in ω to fulfill all conditions on ω. The diagnosis system using all minimal rejectable model shown in (7.21) is an example of a sound and complete diagnosis system with only 5 tests.

Note also that Theorem 7.3 can be used to find a minimum number of models such that there exists a sound diagnosis system for a subset of system behavioral modes. Exchange $\mathbf{B}$ in (7.24) with a set $B \subseteq \mathbf{B}$. If the modified condition (7.24) is fulfilled then the diagnosis system will be sound and complete with respect to the behavioral modes in B but only complete with respect of the behavioral modes not included in B. This can be expressed as

$$\forall z : B \cap C(z) = B \cap D(z) \tag{7.37}$$

and

$$\forall z : (\mathbf{B} \backslash B) \cap D(z) \subseteq (\mathbf{B} \backslash B) \cap C(z) \tag{7.38}$$

Next an example will show the special case when B is the set of all single faults and no-fault. In this example, it is also shown that not only minimal rejectable models can be used to obtain a minimum number of models sufficient for soundness, also larger rejectable models must be used.

Example 7.6

Assume that $B = \{b_1, b_2, b_3, b_4\}$. The minimal number of tests that has to be used is 3. One example is $\omega = \{\{e_3, e_4\}, \{e_1, e_2, e_4\}, \{e_1, e_2, e_3, e_5\}\}$. Note that $\{e_1, e_2, e_3, e_5\}$ is not a minimal rejectable model. If only minimal rejectable models are used, 4 tests are needed. If behavioral models are used, 4 tests are also needed. Assume that $sys = b_2$, that is the first sensor is broken. It has been observed that $z_1 < 0$. The only minimal rejectable model for this z is assumed to be $\{e_1, e_2\}$. This implies that

$$\mathcal{D}(z) = \{b_2, b_5, b_6, b_8\} \tag{7.39}$$

and

$$C(z) = \{b_2, b_5, b_6, b_7, b_8\} \tag{7.40}$$

Expression (7.37) is clearly fulfilled in this example since

$$B \cap C(z) = \{b_2\} = B \cap D(z) \tag{7.41}$$

Furthermore

$$(\mathbf{B} \backslash B) \cap D(z) = \{b_5, b_6, b_8\} \tag{7.42}$$

and

$$(\mathbf{B} \backslash B) \cap C(z) = \{b_5, b_6, b_7, b_8\} \tag{7.43}$$

imply that (7.38) is fulfilled. Note that b_7 is a candidate in (7.43) but not a diagnosis in (7.42). Hence this diagnosis system is not sound with respect to all behavioral modes.

If the sets to design test for is restricted to the minimal rejectable models then an algorithm similar to Algorithm 2 can be defined by using the function `Extend` defined in (3.93). Let Σ be an ordered set enumerating all behavioral modes in $\mathbf{B}$, i.e., $\Sigma = \{\Sigma_{b_1}, \ldots, \Sigma_{b_n}\}$. Then the algorithm can be stated as follows.

Algorithm 9. Ω = `AllMinimalTestSets`(Σ)

```
$\Omega = \emptyset$;
for $i = 1$ to $n$ do
    $\Omega :=$ Extend$(\Omega, \Sigma_{b_i})$;
end for
return $\Omega$
```

This algorithm finds all minimal subsets of ω_m that are solutions to (7.1) for all $b \in \mathbf{B}$. The output of Algorithm 9 when applied to the model in Example 7.5 is $\Omega = \{\omega_m\}$, that is all 5 minimal rejectable models are in this case needed. If $\Sigma = \{\Sigma_b | b \in B\}$ where B is defined as in Example 7.6, then the output set $\Omega = \{\{\{e_1, e_2, e_3\}, \{e_1, e_2, e_4\}, \{e_3, e_4\}, \{e_5\}\}\}$. For further discussions about the algorithm see Section 3.9.1.

7.5 Conclusions

In Chapter 2 we showed that one strategy to construct a diagnosis system was to start with a diagnosis model $\mathbb{M}$ and choose a set $\omega = \{M_1, \ldots, M_n\}$ of rejectable models to test. There, it was also shown that a diagnosis system based on ω can be sound and complete if and only if the set ω fulfills (7.1) for all behavioral modes $b \in \mathbf{B}$.

This chapter has presented solutions ω of (7.1) given a diagnosis model $\mathbb{M}$ with general non-linear equations. It is assumed that it is possible to compute all feasible minimal rejectable models in the non-linear model. A key result is that if ω is chosen to be the set of all feasible minimal rejectable models in the diagnosis model $\mathbb{M}$, then ω fulfills (7.1) for all behavioral modes $b \in \mathbf{B}$ according to Corollary 7.1. Furthermore a sufficient and necessary condition on ω to fulfill (7.1) has been given in Theorem 7.3. It has been shown how this result can be used to minimize and select models such that a sound diagnosis system can be obtained. Furthermore, if the solutions to ω in (7.1) are restricted to include only minimal rejectable models, i.e., if $\omega \subseteq \omega_m$, then all such minimal solution sets are obtained by Algorithm 9.

8

F R M U S M

In model based diagnosis, mathematical models are taken as the basis for diagnosis system construction as described in Chapter 2. If the construction of the diagnosis system follows the strategy "starting from the model" described in Section 2.3.3, a major task is to identify rejectable sub-models. In this chapter we attack the problem of identifying such sub-models in general non-linear dynamic models.

The problem of identifying rejectable models, i.e., models with analytical redundancy is closely related to the problem of variable elimination of the unknowns. For static polynomial models, variable elimination can be done by using gröbner basis techniques (Buchberger, 1965; Cox et al., 1997). For differential algebraic models, differential gröbner basis (Mansfield, 1991) and characteristic sets (Ritt, 1950) techniques provide automatic elimination methods (Wittkopf, 2004; Mishra, 1993). These algorithms will theoretically terminate with the desired output, but often for individual problems the computations either take an unreasonable time to complete, or require more memory than is available (Wittkopf and Reid, 2001). Furthermore, not all non-linear models that we consider here are differential polynomials, e.g. the models can also contain look-up tables.

To handle non-polynomial differential algebraic models and to cope with the computational complexity of identifying model redundancy in polynomial differential algebraic models, *structural analysis* will be used to investigate model redundancy by means efficient graph-based tools (Cassar and Staroswiecki, 1997; Blanke et al., 2003).

The structure of matrices and models are conveniently represented as graphs. Here a bipartite graph will be used to represent the information about which variables that are included in each equation and numerical values and analytical expressions are thereby ignored. The task of finding redundancy in a model can be reformulated as a graph-theoretical problem that can be solved with efficient methods developed for bipartite graphs. These methods are free from numerical problems and have in general lower computational complexity than algebraic elimination methods. On the other hand, the structural approach has its limitations. Ignoring analytical expressions and numerical values, the structural approach gives answers in the generic case. However, for a particular model, it is not sure that the structural approach can identify exactly all rejectable models, if numerical cancellation occurs.

Systematic structural approaches to find rejectable sub-models have been suggested in e.g. (Blanke et al., 2003), (Cassar and Staroswiecki, 1997), (Pulido and Alonso, 2002) (Travé-Massuyès et al., 2001), and (Krysander and Nyberg, 2002a). All these approaches have in common that rejectable models are found among the sub-models with more equations than unknowns. Furthermore, of all these models, it is the minimal ones that have been used to derive analytical redundancy relations. In this chapter, we will formally derive these structural model properties corresponding analytical model properties defined for linear dynamic models.

We start to recapitulate basic graph-theoretical concepts in Section 8.1. Graph-theoretical properties are then associated with matrices of linear dynamic models in Section 8.2 by following the presentation given by (Murota, 2000). A key property used for defining redundancy in linear models is the rank of matrices, and a structural property corresponding to the rank of matrices is also introduced. Based on this structural variant of rank, structural properties corresponding to redundancy, overdetermined set, PO set, and MO set are then derived in Section 8.3. In Section 8.4, the structural characterization of MO sets, i.e., *minimal structurally overdetermined* (MSO) sets is used to derive the structural correspondence to Corollary 6.1. That is, in the generic case soundness can be obtained if all MSO sets are tested.

Then, the structural approach is extended to non-linear dynamic models. For dynamic models, there are different types of bipartite graph representations and these are presented and discussed in Section 8.5. Section 8.6 presents some non-linear dynamic examples which show that models with redundancy can be identified with the proposed graph theoretical methods. Especially, the strategy of testing the MSO sets is extended to non-linear dynamic models. Finally, the conclusions are drawn in Section 8.7.

8.1 Some Basic Graph Theoretical Concepts

Graphs will later be shown to be suitable representations of the structure of matrices and models. Graph-theoretical results lead to efficient algorithms for finding rejectable models. In this section some basic graph theoretical concepts, that can be found in e.g. (Gross and Yellen, 2003; Asratian et al., 1998; Harary, 1969), are recapitulated. Readers familiar with basic graph theory might omit this section.

A *graph* $G = (V, \bar{A})$ consists of two sets. The elements of V are *vertices* and the elements of $\bar{A}$ are *edges*. Vertices are also called *nodes* and edges are sometimes called *arcs*. Each edge has a set of one or two vertices associated to it, which are called *endpoints*. An *edge* is said to *join* its endpoints. A vertex v is *adjacent* to a vertex u if they are joined by an edge. An edge a is *adjacent* to an edge b if they have a common endpoint. A graph is *bipartite*, if its vertices can be partitioned into two sets in such way, that no edge joins two vertices from the same set. A bipartite graph with vertices partitioned into the sets U and V and edges $\bar{A}$ is written $G = (U \cup V, \bar{A})$. Let the two vertex sets be explicitly ordered, lets say $U = \{u_1, u_2, \ldots, u_m\}$ and $V = \{v_1, v_2, \ldots, v_n\}$. Then a *biadjacency matrix* for a bipartite graph G is the $m \times n$ matrix A defined by

$$A_{ij} = \begin{cases} 1 & \text{if } u_i \text{ and } v_j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{8.1}$$

8.1.1 Matchings

A *matching* of $G = (V, \bar{A})$ is a set $\Gamma \subseteq \bar{A}$ of pairwise non-adjacent edges. The size (or cardinality) of a matching Γ is the number of edges in Γ, written $|\Gamma|$. A *maximum size matching* of G is a matching Γ of G having the largest size $|\Gamma|$ of any matching of G. Such matching is also called a *maximum matching* of G or a *maximal matching* of G. Given a bipartite graph $G = (U \cup V, \bar{A})$, a *complete matching* Γ of V into U is a matching such that all vertices in V is an endpoint of an edge in Γ. A matching of G can equally well be a complete matching of U into V. A matching of G that is both a complete matching of U into V and a complete matching of V into U is a *perfect matching* of G. A *path* on a graph $G = (V, \bar{A})$ is a sequence of vertices $v_1, v_2, \ldots, v_n$ such that $(v_i, v_{i+1}) \in \bar{A}$ for all $i \in \{1, \ldots n-1\}$, $v_i \in V$ for all $i \in \{1, \ldots, n\}$, and $v_i \neq v_j$ if $i \neq j$. An *alternating path* is a path in which the edges belong alternately to a matching an not to the matching. A vertex is *free*, if it is not an endpoint of an edge in a matching.

8.2 Structured Matrices

In this section, we will follow the approach presented in (Murota, 2000) and show that the rank of matrices can, in the generic case, be formulated as a maximal matching problem in a bipartite graph. Such matching can be computed efficiently in polynomial time (Asratian et al., 1998; Alt et al., 1991). In the graph theoretical structural approach to find rejectable models in a linear dynamic model we extract the information about which matrix entries that are non-zero, ignoring the numerical values of the entries. Let the degree of a polynomial $b(p)$ be denoted by $\deg(b(p))$.

Definition 8.1 (Structured Matrix). *Given a polynomial matrix $A(p) = [A_{ij}]$, the matrix defined by*

$$(A_{str})_{ij} = \begin{cases} a_{ij} p^{deg(A_{ij})} & \text{if } A_{ij} \neq 0 \\ 0 & \text{if } A_{ij} = 0 \end{cases} \tag{8.2}$$

where a_{ij} are assumed to be algebraically independent parameters *(Murota, 2000) is a* ***structured matrix*** *associated with $A(p)$.*[1]

Note that two different matrices $A(p)$ and $A'(p)$ can be associated with the same structured matrix $A_{str}(p)$. This means that a structured matrix is associated

[1] For our purpose we will not use the degree of the entries, but for consistency with (Murota, 2000) they will be included.

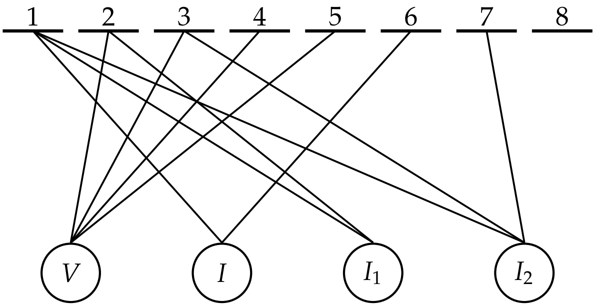

Figure 8.1: The bipartite graph associated with $H(p)$ in (6.65).

with a family of matrices that have a common structure with respect to the non-zero entries and the degrees of the entries.

Example 8.1
Associated with for example the matrix $H(p)$ in (6.65), we consider

$$H_{str}(p) = \begin{bmatrix} 0 & h_1 & h_2 & h_3 \\ h_4 & 0 & h_5 & 0 \\ h_6 & 0 & 0 & p\,h_7 \\ h_8 & 0 & 0 & 0 \\ h_9 & 0 & 0 & 0 \\ 0 & h_{10} & 0 & 0 \\ 0 & 0 & 0 & h_{11} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{8.3}$$

where $h_1, \ldots, h_{11}$ are considered to be algebraically independent parameters.

8.2.1 Structured Matrix Representations

For a matrix $A(p)$ with a row index set R and column index set C, consider the bipartite graph $G = (R \cup C, \bar{A})$ where the edge set $\bar{A}$ of $A(p)$ is defined by

$$\bar{A} = \{(i, j) | i \in R, j \in C, A_{ij} \neq 0\} \tag{8.4}$$

That is, an edge represents a non-zero entry of $A(p)$. To present the structure of the matrix $A(p)$ the corresponding bipartite graph can be used, either as the graph itself or represented with its biadjacency matrix. For easier comprehension of the biadjacency matrix, the zeros will be left out and the ones are marked with an X.

Example 8.2
As an example, consider the matrix $H(p)$ in (6.65). The bipartite graph associated with $H(p)$ is shown in Figure 8.1. The equations are shown as bars and variables

as circles. The biadjacency matrix of this bipartite graph is

$$\begin{array}{c|cccc|} & V & I & I_1 & I_2 \\ 1 & & X & X & X \\ 2 & X & & X & \\ 3 & X & & & X \\ 4 & X & & & \\ 5 & X & & & \\ 6 & & X & & \\ 7 & & & & X \\ 8 & & & & \end{array} \tag{8.5}$$

Note that the structured matrix $H_{str}(p)$ in (8.3) is associated with the same bipartite graph as $H(p)$.

8.2.2 Rank of Structured Matrices

We refer to the rank of $A_{str}(p)$ as the *structural rank*[2] of $A(p)$ and denote it by s-rank $(A(p))$, that means

$$\text{s-rank}\,(A(p)) = \text{rank}\,(A_{str}(p)) \tag{8.6}$$

There is no guarantee that the structural rank coincides with the true rank for a particular and numerically specified matrix $A(p)$. However it is true that rank $(A'(p))$ = s-rank $(A'(p))$ for almost all $A'(p)$ with the same structure, in the sense that $A'_{str}(p) = A_{str}(p)$.

The structural rank has the advantage that it can be computed in an efficient combinatorial way, free from numerical difficulties. This is based on the close relationship, that will be explained next, between sub-determinants of a structured matrix and matchings in the corresponding bipartite graph.

8.2.3 Graph-theoretical Characterization of Structural Rank

The structural rank can as said before be formulated as a graph theoretical property. Next we show that the structural rank of a matrix $A(p)$ is equal to the maximum size of a matching in its associated bipartite graph. To do this let the size of a maximum matching in a graph G be denoted by $\nu(G)$.

Theorem 8.1 ((Murota, 2000)). *The structural rank of a polynomial matrix $A(p)$ is equal to the cardinality of a maximum matching of the associated bipartite graph G, i.e.,*

$$\text{s-rank}\,(A(p)) = \nu(G) \tag{8.7}$$

Proof. For a square $n \times n$ polynomial matrix $A(p)$, its *determinant*, denoted by $\det A(p)$, is defined by

$$\det A(p) = \sum_{\pi \in \mathcal{S}_n} \text{sgn}\,\pi \cdot \prod_{i=1}^{n} A_{i\pi(i)} \tag{8.8}$$

[2] The notions of term rank and generic rank in (Murota, 2000) are both equivalent to structural rank for the models considered in this thesis.

where $\mathcal{S}_n$ denotes the set of all permutations of order n, and $\operatorname{sgn} \pi = \pm 1$ is the signature of the permutation π. A matrix is *nonsingular* if it is square and its determinant is non-zero.

The rank of a matrix is equal to the maximum size of a nonsingular submatrix, i.e.,

$$\operatorname{rank}\,(A(p)) = \max\left\{ |I| \,\middle|\, A[I, J] \text{ is nonsingular, } I \subseteq R, J \subseteq C \right\} \tag{8.9}$$

To determine the structural rank of $A(p)$, (8.6) and (8.9) are combined and we get

$$\begin{aligned} \text{s-rank}\,(A(p)) = \operatorname{rank}\,(A_{str}(p)) = &\\ \max\left\{ |I| \,\middle|\, A_{str}[I, J] \text{ is nonsingular, } I \subseteq R, J \subseteq C \right\} & \qquad (8.10) \end{aligned}$$

The matrix $A_{str}[I, J]$ in (8.10) is nonsingular, by definition, if $\det A_{str}[I, J] \neq 0$. The determinant is only defined for matrices where $|I| = |J|$. Consider an arbitrary $I \subseteq R$ and $J \subseteq C$ such that $|I| = |J| = k$. Then the determinant expansion becomes

$$\begin{aligned} \det A_{str}[I, J] = \sum_{\pi \in \mathcal{S}_k} \operatorname{sgn} \pi \cdot \prod_{i=1}^{k} (A_{str})_{I(i)J(\pi(i))} = &\\ \sum_{\pi \in \mathcal{S}_k} \operatorname{sgn} \pi \cdot \prod_{i=1}^{k} a_{I(i)J(\pi(i))}\, p^{\sum_{i=1}^{k} \deg(A_{I(i)J(\pi(i))})} & \qquad (8.11) \end{aligned}$$

Note that a nonzero term in the expansion corresponds to matchings of size k. Furthermore, there is no cancellation of among different nonzero terms in this expansion due to the independence of the nonzero parameters in $A_{str}(p)$. These two facts imply that the structured matrix $A_{str}[I, J]$ is nonsingular, i.e., $\det A_{str}[I, J] \neq 0$, if and only if there exists a $\pi \in \mathcal{S}_k$ such that

$$\prod_{i=1}^{k} A_{I(i)J(\pi(i))} \neq 0 \tag{8.12}$$

That is, there exists a perfect matching of the bipartite graph associated with the matrix $A[I, J]$ with size k. By applying this arguments to all sub-determinants, it follows that the structural rank of $A(p)$ equals the maximum number k such that $A_{i_1 j_1} \neq 0$, $A_{i_2 j_2} \neq 0$, ..., $A_{i_k j_k} \neq 0$, where i_1, i_2, ..., i_k are distinct rows and j_1, j_2, ..., j_k are distinct columns. The set of pairs (i_1, j_1), (i_2, j_2), ..., (i_k, j_k) defines a maximum size matching of the bipartite graph associated with $A(p)$. This completes the proof. □

Example 8.3

To give an example of how this theorem can be used to compute the structural rank, consider the graph in Figure 8.1 and the structured matrix (8.3) associated with the matrix $H(p)$ given in (6.65). An example of a maximum size matching Γ in the associated bipartite graph consists of the 4 edges $(1, I)$, $(2, I_1)$, $(3, I_2)$, and $(4, V)$. The cardinality of the matching is 4 and the structural redundancy is then also 4 according to Theorem 8.1. Note that the structural rank is, in this case, equal to the rank of the $H(p)$.

The structural rank was equal to the rank in the example. In general, the following is true. Non-singularity for a matrix $A(p)$ implies non-singularity for the matrix $A_{str}(p)$. This fact, (8.9), and (8.10) imply the following theorem.

Theorem 8.2 ((Murota, 2000)). *For any matrix $A(p)$ it holds that*

$$\text{rank}\,(A(p)) \leq \text{s-rank}\,(A(p)) \tag{8.13}$$

Example 8.4
To give an example of a matrix where equality in (8.13) does not hold, consider

$$H = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 2 \end{bmatrix} \tag{8.14}$$

This matrix has structural rank 3, but rank 2. This follows from the fact that $H[\{1,2\},\{1,2\}]$ is rank deficient.

8.2.4 General Matrix Rank Assumption

We are often interested in the case when the structural rank and the rank coincide and therefore we will sometimes refer to the following assumption.

Assumption 8.1. *Given a polynomial matrix $A(p)$ with row set R and column set C, it is assumed that*

$$\text{s-rank}\,(A[I,J]) = \text{rank}\,(A[I,J])$$

for all $I \subseteq R$ and for all $J \subseteq C$.

Under this assumption the matrix $A(p)$ can itself be considered to be a structured matrix, i.e., $A(p)$ enjoys all properties structured matrices have. An example of a model that fulfills the assumption is the $H(p)$ matrix in (6.65) of the electrical circuit. Note that Assumption 8.1 is an assumption of a model of the system not an assumption on the system itself. For the system in question there can be many different equivalent models, some of them may satisfy the assumption and others may not. This is discussed (Murota, 2000) and here it will be illustrated by the following example.

Example 8.5
Consider the a linear static model where H is the matrix defined in (8.14) and L is

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8.15}$$

From Example 8.4, it follows that H does not fulfill Assumption 8.1. By multiplying row one with -1 and add the result to the second row the equivalent model

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 2 \end{bmatrix} x + \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} z = 0 \tag{8.16}$$

is obtained. Both these matrices fulfill Assumption 8.1. Hence, for this example there exist both models that satisfy Assumption 8.1 and others that do not.

8.3 Structural Model Properties

Structural analysis will, as said in the introduction, be used to identify rejectable models, or for linear models overdetermined sets, for test construction. In Chapter 6, redundancy, overdetermined set, PO set, and MO set were defined for linear dynamic models in the form

$$H(p)\,x + L(p)\,z = 0 \tag{8.17}$$

A key property used for defining redundancy in linear models was the rank of matrices, and in the previous section structural rank was introduced as the structural correspondence to the rank of matrices. Based on structural rank, structural model properties will be defined such that they correspond to the analytical model properties in the sense that the structural and analytical notions are equivalent for models with structured matrices. Here we refer to models with structured matrices, but it is also possible to refer to models where the matrix $[H(p)\,L(p)]$ fulfill Assumption 8.1 instead. The definitions will be formulated in general graph theoretical terms to be applicable also for non-linear models later in Sections 8.5-8.6.

All model properties defined in Chapter 6 are based on redundancy. Therefore, we start to define structural redundancy. To get the definition of structural redundancy applicable to any model with its structure given by a bipartite graph, some notation is needed.

For a model $\boldsymbol{M}$ with variables $\boldsymbol{X} \cup \boldsymbol{Z}$, consider the bipartite graph $G(\boldsymbol{M}, \boldsymbol{X} \cup \boldsymbol{Z}) = (\boldsymbol{X} \cup \boldsymbol{Z}, \bar{A})$, where the edge set $\bar{A}$ is defined by

$$\bar{A} = \{(v, e) | v \in \boldsymbol{X} \cup \boldsymbol{Z}, e \in \boldsymbol{M}, v \text{ contained in } e\} \tag{8.18}$$

That is, an edge represents that a variable is included in an equation. Furthermore, given a set $M \subseteq \boldsymbol{M}$ and a variable set $V \subseteq \boldsymbol{X} \cup \boldsymbol{Z}$, let $G(M, V)$ denote the bipartite subgraph of $G(\boldsymbol{M}, \boldsymbol{X} \cup \boldsymbol{Z})$ induced by the vertex set $M \cup V$.

Now, we are ready to define structural redundancy.

Definition 8.2 (Structural Redundancy)**.** *Given a model $\boldsymbol{M}$, let $\varphi_s : 2^{\boldsymbol{M}} \rightarrow \mathbb{Z}$ be a function from the family of subsets M in the set $\boldsymbol{M}$ defined by*

$$\varphi_s\,(M) = \nu(G(M, \boldsymbol{X} \cup \boldsymbol{Z})) - \nu(G(M, \boldsymbol{X})) \tag{8.19}$$

*This number $\varphi_s\,(M)$ is the **structural redundancy** of M.*

According to the definition, it is clear that $\varphi_s\,(M) \geq 0$ for any model M and that $\varphi_s\,(\emptyset) = 0$. The next theorem motivates the definition.

Theorem 8.3. *Given a linear model M with matrices $H(p)$ and $L(p)$, the structural redundancy is equal to*

$$\varphi_s\,(M) = \text{s-rank}\,([H(p)\,L(p)]) - \text{s-rank}\,(H(p)) \tag{8.20}$$

Proof. The bipartite graph associated with the matrix $[H(p)\,L(p)]$ is equal to $G(\boldsymbol{M}, \boldsymbol{X} \cup \boldsymbol{Z})$ and the graph associated with $H(p)$ is equal to $G(\boldsymbol{M}, \boldsymbol{X})$. Then Theorem 8.1 implies that

$$\text{s-rank}\,([H(p)\,L(p)]) = \nu(G(M, \boldsymbol{X} \cup \boldsymbol{Z}))$$

and

$$\text{s-rank}\,(H(p)) = \nu(G(M, \boldsymbol{X}))$$

By substitution of the structural ranks on the right hand side of (8.20), we get the right hand side of (8.19) which completes the proof. □

Let M be a set of equations with matrices $H(p)$ and $L(p)$. Let $[H_{str}(p)\,L_{str}(p)]$ be a structured matrix associated with $[H(p)\,L(p)]$. Then, the model M_{str} defined by the matrices $H_{str}(p)$ and $L_{str}(p)$ is a *structured model* associated with M.

Theorem 8.4. *If M is a set of equations defined by the matrices $H(p)$ and $L(p)$, then it follows that*

$$\varphi_s\,(M) = \varphi\,(M_{str}) \tag{8.21}$$

Proof. The theorem follows from Definition 6.1, Definition 8.2, and (8.6). □

Example 8.6

To give an example of structural redundancy, consider the no-fault model $M_{\mathbf{NF}} = \{1, \ldots, 7\}$ of (6.65). The structural redundancy is, according to Theorem 8.3, equal to

$$\varphi_s\,(M_{\mathbf{NF}}) = \text{s-rank}\,([H[M_{\mathbf{NF}}]\,L[M_{\mathbf{NF}}]) - \text{s-rank}\,(H[M_{\mathbf{NF}}]) = 7 - 4 = 3 \tag{8.22}$$

For this example, the analytical redundancy is equal to the structural redundancy.

In general, the following inequality between structural and analytical redundancy holds.

Theorem 8.5. *If M is a set of equations defined by the matrices $H(p)$ and $L(p)$ where $[H(p)\,L(p)]$ has full row-rank, then it follows that*

$$\varphi_s\,(M) \leq \varphi\,(M) \tag{8.23}$$

Proof. The structural redundancy $\varphi_s\,(M)$ is given by (8.20) and the analytical redundancy is defined in Definition 6.1. Since $[H(p)\,L(p)]$ has full row-rank, Theorem 8.2 implies that

$$\text{s-rank}\,([H(p)\,L(p)]) = \text{rank}\,([H(p)\,L(p)]) \tag{8.24}$$

and

$$\text{s-rank}\,(H(p)) \geq \text{rank}\,(H(p)) \tag{8.25}$$

From (8.24) and (8.25), we get that

$$\begin{aligned}\varphi_s\,(M) = \text{s-rank}\,([H(p)\,L(p)]) - \text{s-rank}\,(H(p)) \leq \\ \text{rank}\,([H(p)\,L(p)]) - \text{rank}\,(H(p)) = \varphi\,(M)\end{aligned}$$

which completes the proof. □

This theorem states that under the condition that $[H(p)\,L(p)]$ has full row-rank, it follows that if structural redundancy exists then analytical redundancy exists. Generically, the structural and analytical redundancy are equal, but inequality in (8.23) can be obtained as the next example will show.

Example 8.7
Consider the matrix H given in (8.14) and a matrix

$$L = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{8.26}$$

For these matrices, it holds that $[H\,L]$ has full-row rank. The structural rank is

$$\varphi_s = \text{s-rank}\,([H\,L]) - \text{s-rank}\,(H) = 4 - 3 = 1 \tag{8.27}$$

and the analytical redundancy is

$$\varphi = \text{rank}\,([H\,L]) - \text{rank}\,(H) = 4 - 2 = 2 \tag{8.28}$$

i.e., $\varphi > \varphi_s$. The reason for the inequality is again that $H[\{1,2\},\{1,2\}]$ is rank deficient.

For real systems, it may happen that some sub-matrix of $H(p)$ is rank deficient. In (Murota, 2000) an electrical circuit is considered and it is shown that a natural model for this system includes rank deficiencies. Moreover, a method to rewrite the model into an equivalent model without including rank deficiencies is described.

Next, we will exemplify that (8.23) can be false if $[H(p)\,L(p)]$ has not full row-rank. Note that this situation should be considered as an exception, since it is possible to find an equivalent model with full row-rank.

Example 8.8
Consider the matrices

$$H = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \qquad L = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \tag{8.29}$$

In this case s-rank (H) = rank (H) = 2 and

$$\text{rank}\,([H\,L]) = 2 < 3 = \text{s-rank}\,([H\,L]) \tag{8.30}$$

This implies that $\varphi = 0 \leq 1 = \varphi_s$, i.e., (8.23) is not true. The difference in structural and analytical redundancy is caused by the rank deficiency in $[H[\{1,2\},\{2\}]\,L[\{1,2\}]$.

A linear dynamic model M was defined to be overdetermined if and only if M has redundancy, i.e., $\varphi\,(M) > 0$. This is for a structured linear dynamic model M_{str} equivalent to $\varphi_s\,(M_{str}) > 0$ according to (8.21). By using structural redundancy the corresponding structural property can be defined.

Definition 8.3 (Structurally Overdetermined Set)**.** *A set M of equations is a **structurally overdetermined (SO) set** if its structural redundancy is positive, i.e* $\varphi_s\,(M) > 0$.

In the same way, structural properties corresponding to PO set and MO set are defined.

Definition 8.4 (Proper Structurally Overdetermined Set)**.** *A structurally overdetermined set M of equations is a* ***proper structurally overdetermined (PSO) set*** *if*

$$\varphi_s(E) < \varphi_s(M) \tag{8.31}$$

for all $E \subset M$.

Definition 8.5 (Minimal Structurally Overdetermined Set)**.** *An SO set of equations is a* ***minimal structurally overdetermined (MSO) set*** *if no proper subset is an SO set.*

Note that an MSO set is also a PSO set. All analytical model properties are transfered to structural properties according to the following theorem.

Theorem 8.6. *A structured model M_{str} is*

a) an overdetermined set if and only if M_{str} is an SO set.

b) a PO set if and only if M_{str} is a PSO set.

c) an MO set if and only if M_{str} is an MSO set.

Proof. We will only show (a) and the other two statements can be proved analogously. The model M_{str} is overdetermined if and only if $\varphi(M_{str}) > 0$ according to Definition 6.2. Since M_{str} is a structured matrix, it follows from Theorem 8.4 that $\varphi_s(M_{str}) = \varphi(M_{str})$. This implies that a model M_{str} is overdetermined, i.e., $\varphi(M_{str}) > 0$, if and only if $\varphi_s(M_{str}) > 0$. The latter is according to Definition 8.3 equivalent to that M_{str} is an SO set. Hence, M_{str} is an overdetermined set if and only if M_{str} is a SO set and this completes the proof of (a). □

For a model where the corresponding matrix $[H(p)\, L(p)]$ fulfills Assumption 8.1, it follows from this theorem that a sub-model is an MO set if and only if the sub-model is an MSO set. This means that all MO sets can be found by finding all MSO sets. Even if this exact relationship does not hold in general, this relationship is a key property that later will be used for finding rejectable models in non-linear models by using a structural method. Before, we consider the non-linear case, we show that a sound diagnosis system can be based on the MSO sets under Assumption 8.1.

8.4 Structural Conditions for Soundness

In Chapter 6, it was concluded that MO sets were suitable to test to obtain soundness. In this section, we formulate the corresponding structural result.

First of all, it was shown in Corollary 3.3 that it is sufficient to check the consistency of all MO sets in a model M_b. Given a structured behavioral model M_b, it follows that the MO sets $E \subseteq M_b$ are the MSO sets $E \subseteq M_b$. Hence the following result is immediate.

Theorem 8.7. *Given a structured linear dynamic behavioral model M_b, it follows that*

$$\mathcal{O}(M_b) = \cap_{M_i \in \mathcal{M}_{MSO}} \mathcal{O}(M_i) \tag{8.32}$$

where $\mathcal{M}_{MSO}$ is the family of all MSO sets $M \subseteq M_b$.

Proof. This theorem follows directly from Theorem 8.6 (c) and Theorem 6.5. □

To give an example, the no-fault model in (6.65) fulfills Assumption 8.1, i.e., the model can be considered to be a structured model. The result given in Theorem 8.7 is therefore applicable. It holds that $\mathcal{M}_{MSO} = \mathcal{M}_{MO}$ for this example. Hence, a structural method to compute all MSO sets can be used as an alternative to use one of the algorithms presented in Chapter 4 for finding all MO sets.

Next, we will extend the structural approach of identifying models with redundancy to the non-linear case. Different structural representations of non-linear dynamic models are presented in the next section. These representations are then used in Section 8.6 where examples motivate test construction based on MSO sets also in the non-linear dynamic case.

8.5 Structural Representations of Non-linear Models

In this and the following sections, the idea of identifying models with redundancy by using structural methods will be extended to the non-linear dynamic case. In this section, we discuss different structural representations for non-linear models. Then Section 8.6 shows that concepts and theory will be applicable to any of these representations.

As for the linear case the structure of a non-linear model M is represented by a bipartite graph $G(M, X \cup Z)$ with variables $X \cup Z$ and equations M as node sets. Analogously to the linear case, there is an edge connecting an equation $e \in M$ and a variable $x \in X \cup Z$ if x is included in e.

Example 8.9
Consider for example the algebraic system

Equation	Expression
e_1	$e^{x_1} \;=\; a\,u$
e_2	$x_1^2 \;=\; b\,x_2$
e_3	$y \;=\; x_2$

(8.33)

where u and y are known variables, x_1 and x_2 are unknown variables, and a and b are known constants. The structure of the model represented as a bipartite graph is shown in Figure 8.2. Note that the structure of the model does not contain the known parameters. The structure of the model represented as a biadjacency matrix is

Equation	Unknown		Known	
	x_1	x_2	u	y
e_1	X		X	
e_2	X	X		
e_3		X		X

(8.34)

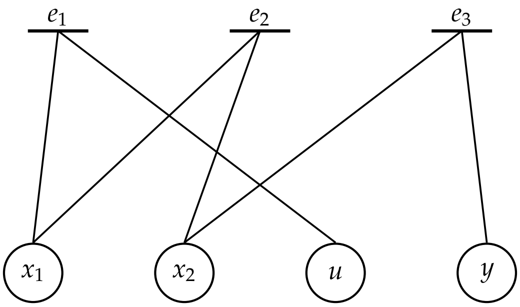

Figure 8.2: Bipartite graph for the model shown in (8.33).

8.5.1 The Structure of Dynamic Models

When considering differential algebraic systems, different alternatives for handling derivatives exist. In this section, three different structural representations of a differential algebraic system are recalled from previous works. These three variants will be exemplified by the following differential algebraic system

Equation	Expression		
e_1	$\dot{x}_1$	$=$	$-x_1^2 + u$
e_2	x_2	$=$	x_1^2
e_3	y	$=$	x_2

(8.35)

where u and y are known, and x_1 and x_2 are unknown signals.

The first structural representation of (8.35) is the following biadjacency matrix of the bipartite graph:

Equation	Unknown		Known	
	x_1	x_2	u	y
e_1	X		X	
e_2	X	X		
e_3		X		X

(8.36)

In this representation all unknowns, i.e., x_1 and x_2, are considered to be signals. There is an "X" in position (i, j) in the biadjacency matrix if x_j or any of its time-derivatives appear in equation e_i. For a linear model with matrices $H(p)$ and $L(p)$, this representation is equal to the bipartite graph associated with the matrix $[H(p)\,L(p)]$. For a general dynamic model with non-zero structural redundancy, differential algebraic elimination methods are needed to eliminate the unknowns. This approach has been used in for example (Frisk, 2001) and (Frisk et al., 2003).

A second structural representation of (8.35) is

Equation	Unknown			Known	
	x_1	$\dot{x}_1$	x_2	u	y
e_1	X	X		X	
e_2	X		X		
e_3			X		X

(8.37)

Unknowns and their time-derivatives are, in contrast to previous representation, considered to be separate independent algebraic variables. Then the equations will no longer represent the same behavior, because there is no guarantee that a signal $\dot{x}$ is the time derivative of x. However, new equations can be obtained by differentiation, so called *prolongation* (Mansfield, 1991), such that the correct relation between variables and their derivatives are implied with these extra equations. For system (8.35), new equations obtained by differentiation are for example

$$\begin{aligned} \dot{e}_2 &: \quad \dot{x}_2 = 2\,x_1\,\dot{x}_1 \\ \dot{e}_3 &: \quad \dot{y} = \dot{x}_2 \end{aligned}$$

Now, with these extra equations the structural representation can be extended

Equation	Unknown				Known		
	x_1	$\dot{x}_1$	x_2	$\dot{x}_2$	u	y	$\dot{y}$
e_1	X	X			X		
e_2	X		X				
$\dot{e}_2$	X	X		X			
e_3			X			X	
$\dot{e}_3$				X			X

(8.38)

The prolongated structure is used in (Krysander and Nyberg, 2002a) and (Krysander and Åslund, 2005). Elimination of the unknowns in models with non-zero structural redundancy in the prolongated structure can, in contrast to the first representation, be done with algebraic methods.

In the third and final structural representation, unknowns and their time-derivatives are, as in the second representation, considered to be separate independent algebraic variables. Thus the equations are purely algebraic and differential equations in the form

$$\dot{x}_i = \frac{d}{dt}x_i$$

are added to relate the variables to its derivatives. The structural representation of (8.35) is

Equation	Unknown			Known
	x_1	$\dot{x}_1$	x_2	
e_1	X	X		
e_2	X		X	
e_3			X	
d	X	X		

(8.39)

where d is the added differential equation. This representation is used for diagnosis in (Blanke et al., 2003).

Now, three different structural representations have been recalled and in Section 8.6 it will be shown that models with redundancy can be found among the SO sets independent of structural representation.

8.6 Finding Rejectable Models by Using Structural Methods

In this section we will exemplify that for any of the structural representation presented in Section 8.5, the structural model properties defined in Section 8.3 are relevant also for finding redundancy in non-linear dynamic models. The idea of testing MSO sets can therefore be extended to the non-linear case.

8.6.1 Structural vs Analytical Redundancy

Structural redundancy and analytical redundancy in a non-linear model is compared in the next example. The structural redundancy is used to define the notions SO set, PSO set, and MSO set and these notions are therefore also compared to analytical redundancy.

Example 8.10
Consider the system of equations (8.35) with the structural representation in (8.36). From Definition 8.2 of structural redundancy, we get

$$\varphi_s(\{e_1, e_2, e_3\}) = 3 - 2 = 1 \tag{8.40}$$

Since $\varphi_s(\{e_1, e_2, e_3\}) > 0$, it follows that $\{e_1, e_2, e_3\}$ is an SO set according to Definition 8.3. In the representation (8.36), different order of derivatives are not distinguished. Since the structural redundancy is non-zero we could expect that $\{e_1, e_2, e_3\}$ contains analytical redundancy. By using a differential algebraic elimination method, the consistency relation

$$\dot{y}^2 - 4y(u - y)^2 = 0 \tag{8.41}$$

can be derived. This means that (8.35) contains redundancy as indicated by its structure.

Now, consider an arbitrary proper subset of $\{e_1, e_2, e_3\}$. The structural redundancy is zero and as indicated by the structural redundancy no consistency relation can be derived from E by differential algebraic elimination tools. However, from the set $\{e_2, e_3\}$ of equations, it is possible to derive that $y \geq 0$. This relation can only be derived with algorithms that handle quantifier elimination (Jirstrand, 1998, 1997) and real algebra, e.g. the function `Reduce` in Mathematica.

To exemplify a PSO set, consider the equations in (8.35) and add a forth equation according to

Equation	Expression		
e_1	$\dot{x}_1$	$=$	$-x_1^2 + u$
e_2	x_2	$=$	x_1^2
e_3	y	$=$	x_2
e_4	z	$=$	$x_2 + x_3^3$

(8.42)

where z is known variable. If the structural representation of the type (8.36) is used, we get

Equation	Unknown			Known		
	x_1	x_2	x_3	u	y	z
e_1	X			X		
e_2	X	X				
e_3		X			X	
e_4		X	X			X

(8.43)

The structural redundancy of $\{e_1, e_2, e_3\}$ and $\{e_1, e_2, e_3, e_4\}$ is one and for all other subsets E of $\{e_1, e_2, e_3, e_4\}$ the structural redundancy is $\varphi_s(E) = 0$. This means that $\{e_1, e_2, e_3\}$ and $\{e_1, e_2, e_3, e_4\}$ are the only SO sets in this model. It is only from these sets the consistency relation (8.41) can be derived. Definition 8.4 implies that the only PSO subset in $\{e_1, e_2, e_3, e_4\}$ is the set $\{e_1, e_2, e_3\}$ and exactly these equations are needed to derive the consistency relation (8.41).

The set $\{e_1, e_2, e_3\}$ is the only MSO set according to Definition 8.6. Previously in Theorem 8.7, we have shown that it is sufficient under some independent assumption to test the consistency of all MSO sets to determine the consistency of all models. Following the same strategy in the non-linear case this corresponds to check the consistency of $\{e_1, e_2, e_3\}$, that is to check the consistency of (8.41). From previous discussion, it follows that this is sufficient for determining the consistency of (8.42).

8.6.2 Test based on MSO Sets

The next example is the water-tank example introduced in Chapter 2 and shows that the tests can be based on MSO sets also in the non-linear dynamic case to obtain a good diagnosis system.

Table 8.1: The structure of the equations of the model in Table 2.2.

Equation	Unknown								Known		
	q_1	w	$\dot{w}$	q_2	f_c	$\dot{f}_c$	f_{yf}	$\dot{f}_{yf}$	u	y_w	y_f
e_1	X								X		
e_2	X		X	X							
e_3		X		X							
e_4		X		X	X						
e_5						X					
e_6		X								X	
e_7				X							X
e_8				X			X				X
e_9								X			

Example 8.11
Consider the water-tank example with the model shown in Table 2.2. Assume that we want to use differential algebraic elimination to construct consistency relations for the MSO sets. Then we will use the structural representation of the type (8.36) shown in Table 8.2. The structure in Table 8.2 contains 12 feasible

MSO sets and these are shown in Table 2.3. These sets are the minimal feasible sets from which consistency relations can be derived by using differential algebraic elimination tools.

Table 8.2: The structure of the equations of the model in Table 2.2.

Equation	Unknown					Known		
	q_1	w	q_2	f_c	f_{yf}	u	y_w	y_f
e_1	X					X		
e_2	X	X	X					
e_3		X	X					
e_4		X	X	X				
e_5				X				
e_6		X					X	
e_7			X					X
e_8			X		X			X
e_9					X			

Finally, we will show that the MSO property can be used to identify minimal rejectable models, independent of structural representation.

Example 8.12
The model (8.35) contains redundancy, because the consistency relation (8.41) can be derived. Three different structural representations for this model have been given in (8.36), (8.38), and (8.39). For each of these three structural representation the sets $\{e_1, e_2, e_3\}$, $\{e_1, e_2, \dot{e}_2, e_3, \dot{e}_3\}$, and $\{e_1, e_2, e_3, d\}$ are MSO sets respectively. Hence, if any one of these representations is used, an MSO set will identify the relevant set of equations corresponding to the consistency relation (8.41).

In the next chapter, we will develop methods for computing MSO sets. Since concept of MSO sets can be used in any of the structural representations presented in Section 8.5, these methods can be applied to any of these structural representations.

8.6.3 Relations to Other Structural Characterizations

In the literature, different structural approaches for finding rejectable models have been proposed. The different approaches use both different structural representations and also different structural characterizations of rejectable models. Since MSO sets can be used in any of the these three structural representations, comparisons to other structural characterizations of rejectable models are possible.

In (Frisk et al., 2003) and (Krysander and Nyberg, 2002a), MSS sets w.r.t. the unknowns are used to find rejectable sub-models. These are defined to be the minimal sets with more equations than unknowns. Later in Theorem 10.3, it will be proven that a set is an MSO set if and only if the set is an MSS w.r.t. the unknowns and contains known variables.

Example 8.13
Consider the MSO set in (8.36). For the set $\{e_1, e_2, e_3\}$, there are 3 equations and 2

unknowns. For any proper subset of $\{e_1, e_2, e_3\}$ there are more or equally many unknowns compared to the number of equations. Hence the MSO set $\{e_1, e_2, e_3\}$ is an MSS set.

In (Blanke et al., 2003), (Izadi-Zamanabadi and Blanke, 2002), and (Travé-Massuyès et al., 2001) *redundancy relations* are used. Given a maximal matching, a free equation vertex is a redundant relation. An MSS set is then given by the equations reached by an alternating path from the redundant equation.

Example 8.14

Consider again the MSO set $\{e_1, e_2, e_3\}$ in (8.36). A maximal matching is $\{(e_1, x_1), (e_2, x_2)\}$. The only free equation is e_3 and this equation is therefore a redundant relation. An alternating path is e_3, x_2, e_2, x_1, and e_1. Hence the set of equations reached by an alternating path is $\{e_1, e_2, e_3\}$. One interpretation of the matching is that e_1 will be used to solve for the variable x_1 and e_2 solves for x_2. Then a computation order is implicitly defined as follows. To be able to compute the value of x_2 from e_2, the value of x_1 must first be computed. Finally, the value of x_2 is inserted in the redundant equation e_3.

In (Pulido and Gonzalez, 2004; Pulido and Alonso, 2002) *evaluation chains* and *minimal evaluation chains* are used. These are equivalent to SO sets and MSO sets respectively.

In conclusion, the names and the definitions of the structural characterizations defer but with minor differences all concepts are equivalent.

8.7 Conclusions

Structural analysis enables one to investigate model redundancy by means efficient graph-based tools. This is useful for identifying models to test. In Theorem 8.4, it is shown that redundancy of a linear model is equal to the structural redundancy in the generic case. Structural redundancy is a graph theoretical property that can be evaluated with efficient methods developed for bipartite graphs. From the definition of structural redundancy, SO, PSO, MSO sets are straightforwardly defined as the structural correspondence to the analytical models properties overdetermined, PO, and MO set respectively. For non-linear models, there is no easy generalization of these analytical model properties. However, the structural model properties can be applied also to non-linear dynamic models, and interpreted in a similar way as for linear systems. We have given examples where the MSO sets are the minimal models that consistency relations can be derived from by elimination tools, i.e., the idea of testing MSO sets can be extended to the non-linear dynamic case.

9

A E A F MSO

In the previous chapter it was shown that a structural approach can be used to identify rejectable sub-models in a large differential-algebraic model, by identifying MSO sets in a structural representation of the model. Several algorithms for computing all MSO sets have been proposed in (Pulido and Gonzalez, 2004; Pulido and Alonso, 2002), (Krysander and Nyberg, 2002a), and (Dustegör et al., 2006; Blanke et al., 2003). However we will show that all these algorithms run into complexity problems when considering large examples.

In this chapter we present a new algorithm, for computing all MSO sets in a structural model. The new algorithm is a structural correspondence of Algorithm 4 for finding all MO sets. This means that if all feasible MSO sets in a diagnosis model is to be computed, then Algorithm 7 can be used but instead of calling Algorithm 4, the new structural algorithm is called.

In all three structural representations of differential-algebraic systems described in Section 8.5, it has been shown in Section 8.6 that MSO sets should be found. The new algorithm will therefore be applicable to all three structural representations.

An introduction to the new algorithm is given in Section 9.1. The introduction includes a brief description of the main idea of the algorithm and a presentation of some existing graph theoretical tools needed for the construction of the algorithm. Section 9.2 gives a theoretical foundation for the algorithm. In Section 9.3 a basic algorithm for finding all MSO sets will be presented and then in Section 9.4 further improvements are described. Then the computational complexity of the proposed algorithm is discussed in Section 9.5. Different algorithms for finding all MSO sets have been presented in previous literature. These are recalled and the complexity of the previous algorithms will be analyzed in Section 9.6. Finally, in Section 9.7, it is shown that the computation time for finding all MSO sets in a Scania truck engine model is significantly decreased by using the new algorithm compared to a previous algorithm.

9.1 Introduction to the New Algorithm

Similar to the analytical algorithm presented in Chapter 4, the basic idea is to find MSO sets in a PSO set by computing a sequence of PSO subsets with decreasing structural redundancy until an MSO set is found.

Example 9.1
For the electrical circuit example presented in Section 2.6, the PO sets shown in Figure 2.2 are equal to the PSO sets. The MSO set $\{4, 5\}$ can be obtained by starting with the PSO set $\{1, 2, 3, 4, 5, 6, 7\}$ with structural redundancy 3 and then compute the PSO subset $\{3, 4, 5, 7\}$ with structural redundancy 2 and finally compute the MSO set $\{4, 5\}$.

This motivates to study the following problem. Given a PSO set M' how should we compute a PSO sets $E \subset M'$ with redundancy $\varphi_s(M') - 1$. Such a computation can be divided into the following two steps:

a) Remove an arbitrary equation $e \in M'$ from M' and let $M = M' \setminus \{e\}$.

b) Find a PSO set $E \subseteq M$ with maximal structural redundancy among the subsets $E' \subseteq M$.

It will be shown in Theorem 9.4 that, given a natural structural assumption that will be presented in Section 9.1.1, the set M defined in (a) has structural redundancy

$$\varphi_s(M) = \varphi_s(M') - 1 \tag{9.1}$$

Furthermore, it will be shown in Theorem 9.3 that there exists a unique PSO set E in M with maximum structural redundancy

$$\varphi_s(E) = \varphi_s(M) = \varphi_s(M') - 1 \tag{9.2}$$

Hence, given a PSO set M' and an equation $e \in M'$, there will be a unique PSO set $E \subseteq M' \setminus \{e\}$ with structural redundancy $\varphi_s(E) = \varphi_s(M') - 1$.

9.1.1 Structural Assumption

In this section a structural assumption is given that implies uniqueness of a PSO set with maximum structural redundancy among all subsets of a model M. In the analytical case, it was necessary to assume that $[H\,L]$ has full row rank to prove the uniqueness. In the structural case, a corresponding assumption is needed, i.e., $[H\,L]$ has full structural rank. From Theorem 8.1 we get the corresponding bipartite graph formulation:

Assumption 9.1. *Given a set M, the graph $G(M, \mathbf{X} \cup \mathbf{Z})$ contains a complete matching of M into $\mathbf{X} \cup \mathbf{Z}$.*

In the remaining part of this chapter, we will, if not otherwise mentioned, always assume that Assumption 9.1 holds. The assumption is easily verified by computing a maximal matching of $G(M, \mathbf{X} \cup \mathbf{Z})$ and check if

$$\nu(G(M, \mathbf{X} \cup \mathbf{Z})) = |M| \tag{9.3}$$

holds true.

A consequence of Assumption 9.1 is that for any set $M' \subseteq M$ with bipartite graph $G(M', \boldsymbol{X} \cup \boldsymbol{Z})$, there exists a complete matching of M' into $\boldsymbol{X} \cup \boldsymbol{Z}$, i.e.,

$$\nu(G(M', \boldsymbol{X} \cup \boldsymbol{Z})) = |M'| \tag{9.4}$$

The structural redundancy is then given by the following theorem.

Theorem 9.1. *Given a set M_b that satisfy Assumption 9.1, the structural redundancy is*

$$\varphi_s(M) = |M| - \nu(G(M, \boldsymbol{X})) \tag{9.5}$$

for any $M \subseteq M_b$.

Proof. The structural redundancy of M is defined by (8.19). Assumption 9.1 implies (9.4) and by substitution of $\nu(G(M, \boldsymbol{X} \cup \boldsymbol{Z}))$ for $|M|$, we get (9.5). □

Since we in the continuation always assume that Assumption 9.1 holds, it follows that the structural redundancy is given by (9.5). The structural redundancy is then only dependent on the graph $G(M, \boldsymbol{X})$ and this will be used in the algorithm.

9.1.2 Computing the PSO Set with Maximum Redundancy

Efficient methods for computing the unique PSO sets with maximum structural redundancy in $G(M, \boldsymbol{X})$ will be based on the *Dulmage-Mendelsohn canonical decomposition* (Murota, 2000; Dulmage and Mendelsohn, 1958). The biadjacency matrix in Figure 9.1 shows a Dulmage-Mendelsohn canonical decomposition of a bipartite graph $G(M, \boldsymbol{X})$. The grey-shaded areas contain ones and zeros, while the white areas only contain zeros. The thick line represents a maximal matching in the graph $G(M, \boldsymbol{X})$ where the rows and columns are rearranged. The model M is partitioned three parts, M^-, M^0, and M^+ and the unknowns are partitioned correspondingly. The set M^+ is exactly the set of equation vertices $e \in M$ such that for any maximum size matching there exists an alternating path between at least one free equation vertex and e. The set $\boldsymbol{X}^+$ is the set of vertices adjacent to at least one vertex in M^+. The set $\boldsymbol{X}^-$ is exactly the set of variable vertices $x \in \boldsymbol{X}$ such that for any maximum size matching there exists an alternating path between at least one free variable vertex and x. The set M^- is the set of vertices adjacent to at least one vertex in $\boldsymbol{X}^-$. The remaining sets of vertices in M and $\boldsymbol{X}$ are M^0 and $\boldsymbol{X}^0$ respectively.

For the three blocks in the decomposition, it holds that $|M^+| > |\boldsymbol{X}^+|$ if $M^+ \neq \emptyset$, $|M^0| = |\boldsymbol{X}^0|$, and $|M^-| < |\boldsymbol{X}^-|$ if $M^- \neq \emptyset$. Therefore, these parts are sometimes called the *over-constrained part*, the *just-constrained part*, and the *under-constrained part* respectively (Blanke et al., 2003). The part denoted M^+ will here be called the *proper structurally overdetermined (PSO) part* of M. We will later show that the PSO part M^+ of M is equal to the PSO subset of M with maximum structural redundancy. Then the step (b) in the beginning of Section 9.1 can be reformulated as

b) Compute the PSO part M^+.

The PSO part M^+ of M can effectively be computed by directly using the definition of M^+.

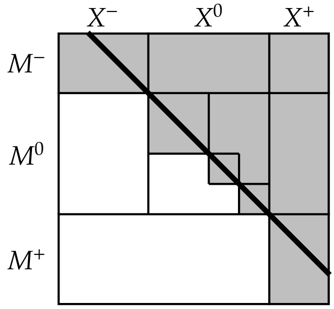

Figure 9.1: The Dulmage-Mendelsohn decomposition of a model M.

In the following example, we illustrate a Dulmage-Mendelsohn canonical decomposition of a bipartite graph and also give some intuition for the different parts in the decomposition when the graph represents the structure of a model.

Example 9.2
Consider the equations in (8.33) and add a forth and a fifth equation according to

Equation	Expression		
e_1	e^{x_1}	$=$	$a\,u$
e_2	x_1^2	$=$	$b\,x_2$
e_3	y	$=$	x_2
e_4	z	$=$	$x_2 + x_3^2$
e_5	x_3	$=$	$x_4 + x_5$

(9.6)

where z is known variable. A structural representation of (9.6) is

Equation	Unknown					Known		
	x_1	x_2	x_3	x_4	x_5	u	y	z
e_1	X					X		
e_2	X	X						
e_3		X					X	
e_4		X	X					X
e_5			X	X	X			

(9.7)

By rearranging the rows and columns, a structure corresponding to Figure 9.1 is

Equation	Unknown					Known		
	x_4	x_5	x_3	x_1	x_2	u	y	z
e_5	X	**X**	X					
e_4			**X**		X			X
e_1				**X**		X		
e_2				X	**X**			
e_3					X		X	

(9.8)

The bold $\boldsymbol{X}$:s denote a maximum matching in $G(M, \boldsymbol{X})$. The sets in Figure 9.1 are for this example $M^- = \{e_5\}$, $M^0 = \{e_4\}$, $M^+ = \{e_1, e_2, e_3\}$, $X^- = \{x_4, x_5\}$, $X^0 = \{x_3\}$, and $X^+ = \{x_1, x_2\}$. The only consistency relation that can be derived from (9.6) by algebraic elimination and is

$$(e^{\sqrt{by}} - au)(e^{-\sqrt{by}} - au) = 0$$

This consistency relation can be obtained by eliminating the variables $X^+ = \{x_1, x_2\}$ in the PSO part $M^+ = \{e_1, e_2, e_3\}$. The value of variable in X^0 can for a given observation be determined by using the equations $M^+ \cup M^0$. Finally, the variables in X^- cannot be uniquely determined for any observation.

In the next section, it will be proven that there exists a unique PSO set with maximal structural redundancy in a model M with $\varphi_s(M) > 0$ and that this PSO set is equal to the PSO part M^+. Furthermore, the theoretical foundation of this chapter is given. For readers who do not intend to go through all technical details, the next section can be omitted.

9.2 Theoretical Foundation

Some basic structural properties will be defined in this section and some basic results that will be used later are given. The following concepts and their theoretical foundation are given in (Ore, 1956). Remember that $\varphi_s(M)$ in the continuation should be interpreted as (9.5), i.e., Assumption 9.1 is assumed to hold.

9.2.1 Surplus of Equations

Let M be a set of equations, V a set of variables, and $G(M, V)$ a bipartite graph. If $E \subseteq M$ is a set of equations, and $X \subseteq V$ is a set of variables, then let the variables X included in some equation in E be denoted by

$$\text{var}_X(E) := \{x \in X | x \text{ is adjacent to some } e \in E\} \tag{9.9}$$

The number of equations more than unknowns in a set M will be called the surplus of M according to the next definition.

Definition 9.1 (Surplus of a Model)**.** *Let $\bar{\varphi}_s : 2^{\boldsymbol{M}} \rightarrow \mathbb{Z}$ be a function, from the family of subsets M in $\boldsymbol{M}$ to the integer numbers, defined by*

$$\bar{\varphi}_s(M) = |M| - |\text{var}_X(M)| \tag{9.10}$$

*This number $\bar{\varphi}_s(M)$ will be called the **surplus of** M.*

Note that $\bar{\varphi}_s(\emptyset) = 0$. The surplus function $\bar{\varphi}_s$ is a super-modular function on the family of equation subsets in $\boldsymbol{M}$ since

$$\bar{\varphi}_s(M_1 \cup M_2) + \bar{\varphi}_s(M_1 \cap M_2) \geq \bar{\varphi}_s(M_1) + \bar{\varphi}_s(M_2) \tag{9.11}$$

for all $M_1 \subseteq \boldsymbol{M}$ and $M_2 \subseteq \boldsymbol{M}$. A set M is said to be a *minimal set of surplus* $\bar{\varphi}_s(M)$ if

$$\bar{\varphi}_s(E) < \bar{\varphi}_s(M) \tag{9.12}$$

for all $E \subset M$.

9.2.2 Characterizing Structural Redundancy

Let M be an arbitrary subset of $\boldsymbol{M}$. Each subset E of M defines a surplus $\bar{\varphi}_s(E)$ and the structural redundancy $\varphi_s(M)$ of M given by (9.5) is equal to

$$\varphi_s(M) = \max_{E \subseteq M} \bar{\varphi}_s(E) \tag{9.13}$$

It holds that $\varphi_s(M) \geq 0$. The surplus of M is clearly less or equal to the structural redundancy of M, i.e.,

$$\varphi_s(M) \geq \bar{\varphi}_s(M) \tag{9.14}$$

and the structural redundancy φ_s is a super-modular function (Ore, 1956), i.e., φ_s satisfies an inequality of the type (9.11). A set M is said to be a *minimal set of structural redundancy* $\varphi_s(M)$ if

$$\varphi_s(E) < \varphi_s(M) \tag{9.15}$$

for all $E \subset M$. For example, an SO set M is defined to be a PSO set if M is a minimal set of structural redundancy $\varphi_s(M)$.

9.2.3 Characterizing the PSO Part M^+

The PSO part can be characterized according to the following results given in (Ore, 1956).

Theorem 9.2. *Let M be a set of equations. Among all subsets E of M with maximum surplus, i.e.,*

$$\bar{\varphi}_s(E) = \varphi_s(M) \tag{9.16}$$

there exists a unique minimal subset. This set is equal to the PSO part M^+ of M.

Corollary 9.1. *The PSO part M^+ of a set M of equations is equal to the minimal set $E \subseteq M$ such that*

$$\varphi_s(E) = \varphi_s(M) \tag{9.17}$$

This means that the PSO part of a set M of equations is equal to the minimal subset of M with maximum structural redundancy. This implies that M^+ contains all structural redundancy of M, i.e.,

$$\varphi_s(M) = \varphi_s(M^+) \tag{9.18}$$

9.2.4 Characterizing PSO Sets

The following lemma gives different characterizations of a PSO set.

Lemma 9.1. *The following three statements about a set M are equivalent:*

(i) The set M is a PSO set.

(ii) The set $M = M^+$ and $M \neq \emptyset$.

(iii) The set M is a minimal set of surplus $\bar{\varphi}_s(M) > 0$.

Proof. (i) ⇒ (ii). Since M is a minimal set of structural redundancy $\varphi_s(M) > 0$, it follows from Corollary 9.1 that $M = M^+$ and $M \neq \emptyset$.

(ii) ⇒ (iii). Since $M = M^+$ and $M \neq \emptyset$, it follows from Theorem 9.2 that M is a minimal set of surplus $\bar{\varphi}_s(M) > 0$.

(iii) ⇒ (i). The set M is a minimal set of surplus $\bar{\varphi}_s(M) > 0$, i.e., satisfies (9.12) for all $E \subset M$. Let M_1 be an arbitrary proper subset of M. It follows that

$$\varphi_s(M_1) = \max_{E \subseteq M_1} \bar{\varphi}_s(E) < \max_{\bar{E} \subseteq M} \bar{\varphi}_s(\bar{E}) = \varphi_s(M)$$

according to (9.12). Since M_1 is an arbitrary proper subset of M, it follows that M is a minimal set of structural redundancy $\varphi_s(M) = \bar{\varphi}_s(M) > 0$, i.e., M is a PSO set. □

Note that given any set M of equations such that

$$M^+ \neq \emptyset \tag{9.19}$$

it follows from Lemma 9.1 that M^+ is a PSO set.

The next lemma shows that the union of PSO sets is a PSO set.

Lemma 9.2. *Given two PSO sets M_1 and M_2, it follows that $M_1 \cup M_2$ is a PSO set and that*

$$\varphi_s(M_1 \cup M_2) \geq \max(\varphi_s(M_1), \varphi_s(M_2)) \tag{9.20}$$

Equality is obtained if and only if $\varphi_s(M_1) \leq \varphi_s(M_2)$ and $M_1 \subseteq M_2$ or $\varphi_s(M_2) \leq \varphi_s(M_1)$ and $M_2 \subseteq M_1$.

Proof. See Theorem 1.2.1 in (Ore, 1956). □

9.2.5 PSO Part and PSO Set

The next theorem gives the needed correspondence between the PSO part of a model M and the PSO subset of M with maximum structural redundancy.

Theorem 9.3. *The maximum structural redundancy among all subsets of a set M is $\varphi_s(M)$. If $\varphi_s(M) > 0$, then there exists a unique PSO set with structural redundancy $\varphi_s(M)$ and this set is equal to M^+.*

Proof. The structural redundancy can be expressed as (9.13). This shows that $\varphi_s(M)$ is the maximum structural redundancy among all subsets of M.

The set M^+ satisfies (9.18) and if $\varphi_s(M) > 0$, then $M^+ \neq \emptyset$. From Lemma 9.1 it follows that M^+ is a PSO set. Hence M^+ is a PSO set with maximum structural redundancy.

To prove the uniqueness, consider an arbitrary PSO set M_0 with structural redundancy $\varphi_s(M_0) = \varphi_s(M)$. This set is according to (9.13), a minimal set of surplus $\bar{\varphi}_s(M_0) = \varphi_s(M)$. From Theorem 9.2, we then get that $M_0 = M^+$. Hence the uniqueness of the PSO set is proved and the theorem follows. □

9.3 New Algorithm

In this section we will present details of the algorithm introduced in Section 9.1. This algorithm is based on a top-down approach in the sense that we start with the entire model an then reduce the size and the structural redundancy of the model step by step until an MSO model remains. The algorithm is based on Algorithm 4 for finding all MO sets but instead of using analytical computations, corresponding structural methods are applied.

The only analytical computation in Algorithm 4 is the computation of the PO part M^* of M in the lumping. To obtain a structural algorithm the computation of the PO part M^* of M will be replaced by the structural correspondence, i.e., to compute the PSO part M^+ of M instead. However, to make this chapter self-contained, the presentation will here be purely structural and similarities with analytical results are commented on.

The algorithm will be based on the following three theorems.

Theorem 9.4. *If M is a PSO set of equations and $e \in M$, then*

$$\varphi_s(M \setminus \{e\}) = \varphi_s(M) - 1 \tag{9.21}$$

Proof. From the definition of the surplus function $\bar{\varphi}_s$ in (9.10), it follows that

$$\bar{\varphi}_s(M \setminus \{e\}) \geq \bar{\varphi}_s(M) - 1 \tag{9.22}$$

From (9.14), we get that

$$\varphi_s(M \setminus \{e\}) \geq \bar{\varphi}_s(M \setminus \{e\}) \tag{9.23}$$

Since M is a PSO set, it follows according Lemma 9.1 that $\bar{\varphi}_s(M) = \varphi_s(M)$. By using this and (9.23) in (9.22), it follows that

$$\varphi_s(M \setminus \{e\}) \geq \varphi_s(M) - 1 \tag{9.24}$$

Since M is a PSO set and therefore also a PSO part according to Lemma 9.1, Corollary 9.1 states that M is a minimal set of structural redundancy $\varphi_s(M)$, i.e.,

$$\varphi_s(M) > \varphi_s(M \setminus \{e\}) \geq \varphi_s(M) - 1 \tag{9.25}$$

This implies (9.21) which completes the proof. □

This theorem, Corollary 9.1, and (9.19), imply that for any PSO set M' with structural redundancy $\varphi_s(M') > 1$ there exists a proper subset $E \neq \emptyset$ which is a PSO set with structural redundancy $\varphi_s(E) = \varphi_s(M') - 1$.

Theorem 9.5. *The set of equations M is an MSO set if and only if M is a PSO set and $\varphi_s(M) = 1$.*

Proof. Assume that M is an MSO set. The set M is therefore an SO set and it follows that $\varphi_s(M) > 0$. The fact that M is an MSO set implies that no proper subset E of M is an SO set, i.e., $\varphi_s(E) = 0$ for any $E \subset M$. This means that M is a minimal set of structural redundancy $\varphi_s(M)$, i.e., M is a PSO set. Assume that M has structural redundancy $\varphi_s(M) > 1$. Then, it follows from Theorem 9.4 that

$$\varphi_s(M \setminus \{e\}) = \varphi_s(M) - 1 \geq 1 \tag{9.26}$$

This implies that $M \setminus \{e\}$ is an SO set which contradicts that M is an MSO set. Hence $\varphi_s(M) = 1$. Assume that M is a PSO set and that $\varphi_s(M) = 1$. From the definition of PSO set, it follows that all proper subsets $E \subset M$ have $\varphi_s(E) = 0$, i.e., E is not an SO set. Hence M is an MSO set. □

Theorem 9.6. *If M is a set of equations, $E \subseteq M$ is a PSO set, and $e \in M \setminus E$, then*

$$E \subseteq (M \setminus \{e\})^+ \tag{9.27}$$

Proof. Let $M \setminus \{e\}$ be denoted by $\hat{M}$. The fact that $E \subseteq \hat{M}$ implies that $E^+ \cup \hat{M}^+ \subseteq \hat{M}$ and from (9.18) also that

$$\varphi_s(\hat{M}^+ \cup E^+) \leq \varphi_s(\hat{M}) = \varphi_s(\hat{M}^+) \tag{9.28}$$

Lemma 9.2 implies that

$$\varphi_s(\hat{M}^+) \leq \max(\varphi_s(\hat{M}^+), \varphi_s(E^+)) \leq \varphi_s(\hat{M}^+ \cup E^+) \tag{9.29}$$

The inequalities (9.28) and (9.29) give that

$$\varphi_s(\hat{M}^+ \cup E^+) = \varphi_s(\hat{M}^+) \tag{9.30}$$

and $\varphi_s(E^+) \leq \varphi_s(\hat{M}^+)$. This, and the equality in (9.30) imply that

$$E^+ \subseteq \hat{M}^+ \tag{9.31}$$

according to Lemma 9.2. Since E is a PSO set, it follows from Lemma 9.1 that $E^+ = E$. This, (9.31), and that $\hat{M} = M \setminus \{e\}$ imply (9.27) and this completes the proof. □

Theorem 9.4 reveals how the structural redundancy decreases when one equation is removed. It follows from this theorem that if we start with any PSO set of equations we can alternately remove equations and computing the PSO part until the structural redundancy becomes 1. We have then found an MSO-set, according to Theorem 9.5. Finally, Theorem 9.6 implies that an arbitrary MSO set can be obtained recursively this way. By using this principle, the algorithm becomes as follows. The input set M can without loss of generality be assumed to be a PSO set.

Algorithm 10. `FindMSO`(M)

```
if $\varphi_s(M) = 1$ then
    $\mathcal{M}_{MSO} := \{M\}$;
else
    $\mathcal{M}_{MSO} := \emptyset$;
    for each equation $e$ in $M$ do
        $M' := (M \setminus \{e\})^+$;
        $\mathcal{M}_{MSO} := \mathcal{M}_{MSO} \cup$ FindMSO($M'$);
    end for
end if
return $\mathcal{M}_{MSO}$
```

From the discussion above, it follows that the sets found in $\mathcal{M}_{MSO}$ are MSO sets and that all MSO sets are found.

Example 9.3
To illustrate the steps in the algorithm, consider the following PSO set consisting of four equations and two unknown variables:

Equation	Unknown	
	x_1	x_2
e_1	X	
e_2	X	X
e_3		X
e_4		X

(9.32)

The structural redundancy of this set of equations is 2. When entering the algorithm, e_1 is removed and the set M' becomes $(M\backslash\{e_1\})^+ = \{e_3, e_4\}$. In this case $\varphi_s(M') = 1$ and the equation set is saved as an MSO in $\mathcal{M}_{MSO}$. Then e_2 is removed and $M' = (M\backslash\{e_2\})^+ = \{e_3, e_4\}$. This means that the same MSO set is found once again. Next e_3 is removed and the MSO set $\{e_1, e_2, e_4\}$ is found. Finally e_4 is removed and the MSO set $\{e_1, e_2, e_3\}$ is found. Thus, three MSO sets are found and the output of Algorithm 10 is $\mathcal{M}_{MSO} = \{\{e_3, e_4\}, \{e_1, e_2, e_4\}, \{e_1, e_2, e_3\}\}$.

Since the same MSO set $\{e_3, e_4\}$ is found twice, we can suspect that the algorithm is not optimal in terms of efficiency. The next section will therefore present improvements in order to increase the efficiency.

9.4 Improvements

A straightforward improvement is of course to prohibit that any of the MSO sets are found more than once. Another and more sophisticated improvement is that sets of equations can be lumped together in order to reduce the size and the complexity of the structure. The proposed reduction preserves structural redundancy and it is therefore possible to use the reduced structure to find all MSO sets in the original structure.

9.4.1 Structural Reduction

The reduction is based on a new unique decomposition of the PSO part of a bipartite graph. An illustration of the decomposition is shown in Figure 9.2 as a biadjacency matrix. The decomposition can be defined as follows. Let R_s be a relation on the set M of equations defined by $(e', e) \in R_s$ if

$$e' \notin (M \setminus \{e\})^+ \tag{9.33}$$

Note the similarity to (4.18) in the analytical case. Now we show that R_s is an equivalence relation.

Lemma 9.3. *The relation R_s is an equivalence relation on a PSO set M.*

Proof. It follows directly from the definition that R_s is reflexive. If $(e', e) \in R_s$, then it follows from (9.33) and Theorem 9.6, with E replaced by $(M \setminus \{e\})^+$, that $(M \setminus \{e\})^+ \subseteq (M \setminus \{e'\})^+$. Theorem 9.4 and Theorem 9.6 imply that both sets have the same structural redundancy and that $(M \backslash \{e\})^+ = (M \backslash \{e'\})^+$. Hence $(e, e') \in R_s$ and R_s is therefore symmetric. Furthermore if $(e_1, e_2) \in R_s$ and $(e_2, e_3) \in R_s$, then it holds that $(M \setminus \{e_1\})^+ = (M \setminus \{e_2\})^+ = (M \setminus \{e_3\})^+$, which implies that R_s is transitive. The relation R_s is therefore an equivalence relation which completes the proof. □

The set M can then be partitioned into m disjoint equivalence classes M_i. For each equation set M_i, the set X_i is defined as the unknowns only included in M_i and

$$X_0 = X \setminus (\bigcup_{i \neq 0} X_i)$$

The first n equivalence classes in the Figure 9.2 are assumed to be the equivalence classes with cardinality strictly greater than one. The partition has the following property.

Lemma 9.4. *If M is a PSO set, then for all its equivalence classes M_i defined by (9.33), it holds that*

$$|M_i| = |X_i| + 1 \tag{9.34}$$

Proof. Let M_i be an arbitrary equivalence class which according to the decomposition implies that for any $e \in M_i$, $(M \setminus \{e\})^+ = M \setminus M_i$. Then we form

$$\bar{\varphi}_s(M) - \bar{\varphi}_s(M \setminus \{e\})^+ = (|M| - |X|) - (|M \setminus M_i| - |X \setminus X_i|)$$

which can be simplified to

$$\bar{\varphi}_s(M) - \bar{\varphi}_s(M \setminus \{e\})^+ = |M_i| - |X_i|$$

Since M and $(M \setminus \{e\})^+$ are PSO sets, it follows according to Theorem 9.2 that

$$\varphi_s(M) - \varphi_s(M \setminus \{e\})^+ = |M_i| - |X_i|$$

Then Theorem 9.4 and (9.18) imply (9.34). □

It follows from Lemma 9.4, that there is one more equation than unknowns, X_0 excluded, in each block $1 \leq i \leq m$ in Figure 9.2. Furthermore for $n+1 \leq i \leq m$ in the figure, M_i has cardinality 1 and $X_i = \emptyset$.

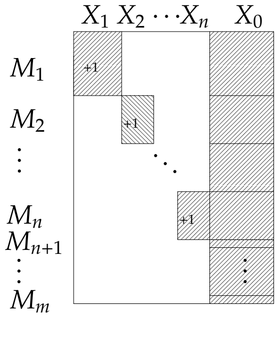

Figure 9.2: A structural decomposition of a PSO set.

By using this partition, all PSO sets can be represented as follows.

Theorem 9.7. *If M and $E \subseteq M$ are two PSO set, then E is a union of equivalence classes defined by (9.33), i.e.,*

$$E = \bigcup_{i \in I} M_i$$

where $I \subseteq \{1, 2, \dots m\}$.

Proof. To prove Theorem 9.7, we will use the following equivalent formulation. For any equivalence class M_i defined by (9.33) such that $E \cap M_i \neq \emptyset$ it follows that $M_i \subseteq E$. Assume that this is false, i.e., there exists an equivalence class M_i such that $E \cap M_i \neq \emptyset$ and $M_i \not\subseteq E$. Then there exists an $e \in M_i \setminus E \subseteq M \setminus E$. From Theorem 9.6, it follows that $E \subseteq (M \setminus \{e\})^+$. This and the definition of M_i imply that $E \subseteq M \setminus M_i$, which contradicts the assumption and the theorem follows. □

A new bipartite graph is formed with equivalence classes $\{M_i\}$ and the unknowns X_0 as node sets. The unknowns connected to M_i are $\text{var}_{X_0}(M_i)$.

Example 9.4
The reduction of (9.32) is

equivalence class M_i	unknown x_2
$\{e_1, e_2\}$	X
$\{e_3\}$	X
$\{e_4\}$	X

and the decomposition is given by $M_1 = \{e_1, e_2\}$, $M_2 = \{e_3\}$, $M_3 = \{e_4\}$, $X_0 = \{x_2\}$, $X_1 = \{x_1\}$, and $X_2 = X_3 = \emptyset$.

Note that it is only equivalence classes of cardinality greater than one that give a reduction. An interpretation of this reduction is that the two first equations are used to eliminate the unknown variable x_1. In the lumped structure, each equivalence class is considered as one equation and the definitions of PSO set, MSO set, and structural redundancy are thereby extended to lumped structures. In the example above we have $\varphi_s(\{\{e_1, e_2\}, \{e_3\}, \{e_4\}\}) = 2$. The structural redundancy for the lumped and the original structure are always the same.

The reduction is justified by the following theorem, which shows that there is a one-to-one correspondence between the PSO sets in the original and in the lumped structure. The reduced structure can therefore be used to find all PSO sets in the original structure.

Theorem 9.8. *The set $\{M_i\}_{i \in I}$ is a PSO set in the lumped structure if and only if $\cup_{i \in I} M_i$ is a PSO set in the original structure.*

To prove Theorem 9.8, the following lemma will be used.

Lemma 9.5. *If M is a PSO set and $\{M_i\}_{i \in I}$ are the equivalence classes induced by R_s, then*

$$\bar{\varphi}_s(\cup_{i \in I'} M_i) = \bar{\varphi}_s(\{M_i\}_{i \in I'}) \tag{9.35}$$

for all $I' \subseteq I$.

Proof. By using the notation of the structural decomposition described in Section 9.4.1 the surplus of $\cup_{i \in I'} M_i$ can be expressed as

$$\bar{\varphi}_s(\cup_{i \in I'} M_i) = |\cup_{i \in I'} M_i| - |\cup_{i \in I'} X_i| - |\text{var}_{X_0}(\cup_{i \in I'} M_i)| \tag{9.36}$$

which can be rewritten as

$$\bar{\varphi}_s\left(\cup_{i\in I'} M_i\right) = \sum_{i\in I'}(|M_i| - |X_i|) - |\mathrm{var}_{X_0}(\cup_{i\in I'} M_i)| \tag{9.37}$$

Lemma 9.4 states that $|M_i| = |X_i| + 1$ for all $i \in I$, and consequently that

$$\bar{\varphi}_s\left(\cup_{i\in I'} M_i\right) = |I'| - |\mathrm{var}_{X_0}(\cup_{i\in I'} M_i)| \tag{9.38}$$

which is equal to $\bar{\varphi}_s\left(\{M_i\}_{i\in I'}\right)$. □

Now, the proof of Theorem 9.8 follows.

Proof. Assume that $\cup_{i\in J} M_i$ is a PSO set. From Lemma 9.1, it follows that

$$\bar{\varphi}_s\left(\cup_{i\in J'} M_i\right) < \bar{\varphi}_s\left(\cup_{i\in J} M_i\right) \tag{9.39}$$

for all $J' \subset J$. From Lemma 9.5, it then follows that

$$\bar{\varphi}_s\left(\{M_i\}_{i\in J'}\right) < \bar{\varphi}_s\left(\{M_i\}_{i\in J}\right) \tag{9.40}$$

for all $J' \subset J$. Hence $\{M_i\}_{i\in J}$ is a minimal set of surplus $\bar{\varphi}_s\left(\{M_i\}_{i\in J}\right)$, i.e., $\{M_i\}_{i\in J}$ is a PSO set according to Lemma 9.1.

Now, we will show the reverse implication. Assume that $\{M_i\}_{i\in J}$ is a PSO set. If $M' \subset \cup_{i\in J} M_i$, then

$$M' \supseteq (M')^+ = \cup_{i\in J'} M_i \tag{9.41}$$

for some $J' \subset J$ according to Theorem 9.7. Since $\{M_i\}_{i\in J}$ is a PSO set, it follows from Lemma 9.1 and Lemma 9.5 that

$$\bar{\varphi}_s\left(\cup_{i\in J} M_i\right) = \bar{\varphi}_s\left(\{M_i\}_{i\in J}\right) > \bar{\varphi}_s\left(\{M_i\}_{i\in J'}\right) = \bar{\varphi}_s\left(\cup_{i\in J'} M_i\right) \tag{9.42}$$

From Theorem 9.2 and (9.41), it follows that

$$\bar{\varphi}_s\left(\cup_{i\in J'} M_i\right) = \bar{\varphi}_s\left(M'\right)^+ \geq \bar{\varphi}_s\left(M'\right) \tag{9.43}$$

The inequalities (9.42) and (9.43) imply that $\cup_{i\in J} M_i$ is a minimal set of surplus $\bar{\varphi}_s\left(\cup_{i\in J} M_i\right)$, i.e., $\cup_{i\in J} M_i$ is a PSO set according to Lemma 9.1. □

9.4.2 Improved Algorithm

A drawback with Algorithm 10, presented in Section 9.3, is that some of the MSO sets are found more than once. There are two reasons why this happens and these can be illustrated using the following example:

Equation	Unknown	
	x_1	x_2
e_1	X	
e_2	X	X
e_3		X
e_4		X
e_5		X

(9.44)

First, the same PSO set $\{e_3, e_4, e_5\}$ is obtained if either e_1 or e_2 is removed. Second, the same MSO set is obtained if the order of equation removal is permuted. For example, the MSO set $\{e_4, e_5\}$ is obtained if first e_1 or e_2 and then e_3 is removed but also if the order of removal is reversed.

To illustrate how these two problems are handled in the next improved algorithm, i.e., Algorithm 11, we use the example (9.44).

To avoid the first problem the lumping described in previous section is used. Initially we start with the set $M = \{e_1, e_2, e_3, e_4, e_5\}$ and e_1 and e_2 are lumped together and the resulting set is $\mathcal{S}' = \{\{e_1, e_2\}, \{e_3\}, \{e_4\}, \{e_5\}\}$. Similar to the basic algorithm we remove one equivalence class at a time from $\mathcal{S}'$ and make a subroutine call which returns all MSO sets in the input set.

To avoid the problem with permuted removal order an additional input set $\mathcal{E}'$ is used which contains the equivalence classes that are allowed to be removed in the recursive calls.

In the example, we start initially with the set $\mathcal{E}' = \mathcal{S}'$, meaning that all equivalence classes are allowed to be removed. In the first step the equivalence class $\{e_1, e_2\}$ is removed and the subroutine is called with the input sets

$$\mathcal{S}' \setminus \{\{e_1, e_2\}\} \text{ and } \mathcal{E}' = \{\{e_3\}, \{e_4\}, \{e_5\}\}$$

To prevent that the order of removal is permuted we remove the equivalence class $\{e_1, e_2\}$ permanently from $\mathcal{E}'$. In the following step the equivalence class $\{e_3\}$ is removed and the inputs are

$$\mathcal{S}' \setminus \{\{e_3\}\} \text{ and } \mathcal{E}' = \{\{e_4\}, \{e_5\}\}$$

Following the same principles, the final calls are made with the input sets

$$\begin{aligned} &\mathcal{S}' \setminus \{\{e_4\}\} \text{ and } \mathcal{E}' = \{\{e_5\}\}, \\ &\mathcal{S}' \setminus \{\{e_5\}\} \text{ and } \mathcal{E}' = \emptyset \end{aligned}$$

To apply these ideas in all steps in the recursive algorithm, the lumping strategy has to be extended to subsets of previously lumped structures. Equivalence classes are then lumped together into new sets of equations by taking the union of the sets in the equivalence class. We illustrate this with a new example. Assume that we start with six equations and that e_2 and e_3 are lumped together and the following structure is obtained:

Equation	Unknown	
	x_1	x_2
$\{e_1\}$	X	
$\{e_2, e_3\}$	X	
$\{e_4\}$	X	X
$\{e_5\}$		X
$\{e_6\}$		X

(9.45)

In the first recursive call $\{e_1\}$ is removed and the graph corresponding to the remaining part has the same structure as in (9.32). Now,

$$[\{e_2, e_3\}]_s = [\{e_4\}]_s = \{\{e_2, e_3\}, \{e_4\}\}$$

where $[E]_s$ denotes the equivalence class containing E. The sets $\{e_2, e_3\}$ and $\{e_4\}$ are therefore lumped together into the set $\{e_2, e_3, e_4\}$.

Given a model $\mathcal{S}$ and corresponding set $\mathcal{E}$, the lumped structure $\mathcal{S}'$ is constructed as described above and a problem is then how to form the new set $\mathcal{E}'$ of equivalence classes that are allowed to be removed in the new structure $\mathcal{S}'$. The following principle will be used. An equivalence class in $\mathcal{S}'$ is allowed to be removed, i.e., belongs to $\mathcal{E}'$, if and only if it is a union of classes that are all allowed to be removed in $\mathcal{S}$, i.e., belongs to $\mathcal{E}$. It will be shown that, in this way, all MSO sets are found once and only once.

It is sufficient to only lump equivalence classes with an non-empty intersection with $\mathcal{E}$ and this is used in the algorithm. To do this partial lumping we will use the notation $\mathcal{S}' = \texttt{Lump}(E, \mathcal{S})$ in the algorithm to denote that only the equivalence class $[E]_s$ in $\mathcal{S}$ is lumped and that the other equations remain unchanged. The improved algorithm can now formally be written as follows.

Algorithm 11. `MSO`(M)

```
$\mathcal{S} := \{\{e\}|e \in M^+\}$;
$\mathcal{M}_{MSO}$ := FindMSO($\mathcal{S}, \mathcal{S}$);
return $\mathcal{M}_{MSO}$;

Subroutine: FindMSO($\mathcal{S}, \mathcal{E}$)
if $\varphi_s(\mathcal{S}) = 1$ then
    $\mathcal{M}_{MSO} := \{\cup_{E \in \mathcal{S}} E\}$;
else
    $\mathcal{E}' := \emptyset; \mathcal{S}' := \mathcal{S}$;
    % Lump the structure $\mathcal{S}'$ and create $\mathcal{E}'$
    while $\mathcal{E} \neq \emptyset$ do
        Select an $E \in \mathcal{E}$;
        $\mathcal{S}'$ := Lump($[E]_s, \mathcal{S}'$);
        if $[E]_s \subseteq \mathcal{E}$ then
            $\mathcal{E}' := \mathcal{E}' \cup \{\cup_{E' \in [E]_s} E'\}$;
        end if
        $\mathcal{E} := \mathcal{E} \setminus [E]_s$;
    end while
    $\mathcal{M}_{MSO} := \emptyset$;
    % Make the recursive calls
    while $\mathcal{E}' \neq \emptyset$ do
        Select an $E \in \mathcal{E}'$;
        $\mathcal{E}' := \mathcal{E}' \setminus \{E\}$;
        $\mathcal{M}_{MSO} := \mathcal{M}_{MSO} \cup$ FindMSO($\mathcal{S}' \setminus \{E\}, \mathcal{E}'$);
    end while
end if
return $\mathcal{M}_{MSO}$
```

The algorithm is justified by the following result.

Theorem 9.9. *If Algorithm 11 is applied to a set M, then each MSO set contained in M is found once and only once.*

Proof. First it is shown that each MSO set is found at least once. Let $E \subseteq M$ be an arbitrary MSO set. A branch, of the recursive tree, that results in this MSO set can be obtained in the following way: In each recursive step, chose the *first* branch where an equivalence class not included in E is removed. It follows from Theorem 9.6 and Theorem 9.8 that by following this branch, a sequence of decreasing PSO sets all containing E is obtained. Hence the MSO set E is found this way.

Finally, it is shown that the same MSO set E can not be found if we deviate from the branch described above, i.e., that the MSO set E is found only once. In each recursive step, in all branches that precede this branch, only equivalence classes contained in E have been removed. Therefore, these branches do not result in the set E. On the other hand all succeeding branches contain the *first* equivalence class $\hat{E}$ not contained in E, i.e., the class removed in the branch that gives the set E. This follows from the fact that $\hat{E}$ has been removed from $\mathcal{E}$ and is not allowed to be removed. Furthermore in all lumped structures in these branches, $\mathcal{E}'$ is constructed such that $\hat{E}$ is an equivalence class not contained in $\mathcal{E}'$. Hence, the branch described above is the only branch that results in the MSO set E. This completes the proof. □

Note that `FindMSO` corresponds to `FindMO` in Algorithm 4. The difference is that we use the structurally defined equivalence relation R_s instead of the analytical equivalence relation R used in Algorithm 4. To exemplify the steps of Algorithm 11, the example in Section 4.6 can be reused. It turns out that Algorithm 4 and Algorithm 11 will traverse exactly the same sets.

9.5 Computational Complexity

The objective of this section is to investigate the computational complexity of Algorithm 11. In general the number of MSO sets may grow exponentially in the number of equations. This gives a lower bound for the computational complexity in the general case. However, in many applications the order of structural redundancy is low and it will be shown that in this case better computational complexity can be achieved. The redundancy is often low due to the fact that the structural redundancy depends on the number of available sensors, which are often expensive. One example of this is given in Section 9.7. In this section the computational complexity of the algorithm will be analyzed in the case where the structural redundancy is low.

The worst case is when all unknown variables are included in each equation. Algorithm 11 traverses the PSO sets exactly once in the subset lattice. The following lemma gives an upper bound for the number of PSO sets.

Lemma 9.6. *Given a model with n equations and with structural redundancy φ, there are at most*

$$\sum_{k=n-\varphi+1}^{n} \binom{n}{k} \tag{9.46}$$

PSO subsets.

Proof. In the worst case the PSO sets are all subsets of equations with cardinality strictly greater than the number of unknowns in the original model, i.e., greater than $n - \varphi$. The number of subsets with k equations is in this case

$$\binom{n}{k}$$

which gives the result in the lemma. □

The next theorem gives the computational complexity in the case of low structural redundancy.

Theorem 9.10. *For a fixed order of structural redundancy φ, Algorithm 11 has order of $n^{\varphi+1.5}$ time complexity, where n is the number of equations.*

Proof. An upper bound for the number of PSO sets is given by Lemma 9.6. For $k \geq n - \varphi + 1$ the terms in (9.46) can be estimated from above as

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \leq \frac{n!}{k!} \leq n^{\varphi-1} \tag{9.47}$$

and the number of terms is fixed. Hence the sum is less than $\varphi\, n^{\varphi-1}$. In the worst case, the number of times the PSO part has to be calculated is given by the sum (9.46). To compute the PSO part has order of $n^{2.5}$ time complexity (Murota, 2000). Hence, Algorithm 11 has order of $n^{\varphi+1.5}$ time complexity. □

9.6 Comparison with Previous Algorithms

Different algorithms for finding MSO sets have been presented in previous literature (Pulido and Gonzalez, 2004; Krysander and Nyberg, 2002a; Blanke et al., 2003; Ploix et al., 2005; Lorentzen et al., 2003). In these works, there are basically three different algorithms for finding all MSO sets. These will now be recalled and the complexity of the previous algorithms will then be analyzed.

9.6.1 Previous Algorithms

The first algorithm for finding all MSO sets was presented in (Pulido and Alonso, 2002) and further developed in (Pulido and Gonzalez, 2004). Independently the same algorithm was presented in (Krysander and Nyberg, 2002a). The basic principle is to choose one equation as the redundant equation and then find all possible ways to compute structurally all unknowns in the redundant equations. The redundant equation is first chosen to be the first equation and then the second and so on until the last equation is the redundant equation. When all possible ways to compute all unknowns in the first equation are found, all MSO sets including the first equation have been found. This means that the first equation will not be used further in the search for more MSO models.

Example 9.5
Consider a PSO set of equations with a structure

Equation	Unknown		
	x_1	x_2	x_3
e_1		X	
e_2	X	X	
e_3		X	X
e_4	X		X
e_5			X

(9.48)

The redundant equations (RE) and matchings computed by the first algorithm is shown in Table 9.1. First, e_1 is chosen to be the redundant equation. Equation e_1 includes the unknown variable x_2. The unknown variable x_2 can be eliminated structurally with e_2 and this is indicated as the matching (x_2, e_2) in the first row (a) of Table 9.1. The only free unknown variable vertex in $G(\{e_1, e_2\}, \boldsymbol{X})$ is x_1 and this variable can be eliminated structurally by using equation e_4. Therefore, the next assignment is (x_1, e_4) in row (a). The only free unknown variable vertex in $G(\{e_1, e_2, e_4\}, \boldsymbol{X})$ is the variable x_3 and this variable can be eliminated structurally by using equation e_3 and the assignment (x_3, e_3) is made. Now, all unknowns in $G(\{e_1, e_2, e_3, e_4\}, \boldsymbol{X})$ are matched and $\{e_1, e_2, e_3, e_4\}$ is an MSO set. Backtracking is then used and the unknown variable x_3 can also be eliminated structurally with e_5 instead of using e_3. This is indicated in the row (b) with (x_3, e_5). All unknowns are matched in $G(\{e_1, e_2, e_4, e_5\}, \boldsymbol{X})$ and $\{e_1, e_2, e_4, e_5\}$ is an MSO set. Continuing in this way Table 9.1 is obtained. By construction, this table corresponds to a depth first search tree where each row in the table corresponds to a leaf node in the search tree.

In the first row, it is seen that the redundant equation e_1 and a matching (x_2, e_2), (x_1, e_4), (x_3, e_3) defines the MSO set $\{e_1, e_2, e_3, e_4\}$. On row (d), we have that the redundant equation e_1 and the matching (x_2, e_3), (x_3, e_5) defines the MSO set $\{e_1, e_3, e_5\}$. Note that this matching is not a complete matching w.r.t. all unknowns, it is only a complete matching of the unknowns included in $G(\{e_1, e_3, e_5\}, \boldsymbol{X})$. The same MSO set can be found several times as seen by comparing row (a) and (c).

The second algorithm for finding all MSO sets was presented in (Blanke et al., 2003). All maximal matchings are first enumerated. Then for each maximal matching and for each free equation for this matching, an MSO set is given by the equations reached by an alternating path from the free equation. For further details see (Dustegör et al., 2006).

Example 9.6
The matchings and the redundant equations computed by the second algorithm when applied to (9.48) is shown in Table 9.2. In the second algorithm, a maximal matching is found before a redundant equation is chosen. Therefore, the order of the columns Matching and RE in Table 9.2 are reversed. The maximal matchings are obtained by first assigning the variables nodes x_1 then x_2 and finally x_3. In Table 9.2, it can be seen that there are 10 different maximal matchings that are complete matchings of the unknowns into the equations.

Table 9.1: Matchings, redundant equations, and MSO sets computed by the first algorithm.

	RE	Matching			MSO Set
a	e_1	(x_2, e_2)	(x_1, e_4)	(x_3, e_3)	$\{e_1, e_2, e_3, e_4\}$
b				(x_3, e_5)	$\{e_1, e_2, e_4, e_5\}$
c		(x_2, e_3)	(x_3, e_4)	(x_1, e_2)	$\{e_1, e_2, e_3, e_4\}$
d			(x_3, e_5)		$\{e_1, e_3, e_5\}$
e	e_2	(x_1, e_4)	(x_2, e_3)	(x_3, e_5)	$\{e_2, e_3, e_4, e_5\}$
f	e_3				
g	e_4				
h	e_5				

Furthermore, for each complete matching there are two possible redundant equations. This means that we find MSO sets 20 times, to find the 4 MSO sets.

The third algorithm for finding all MSO sets is presented in (Ploix et al., 2005). This method is based on elimination rules. The unknowns are eliminated in a specified order. Each unknown is eliminated in all possible ways. For each way, the equations used form an MSO set.

Example 9.7

Consider the structure in (9.48). Assume that the unknowns are eliminated in accordance to their enumeration. The first variable x_1 can only be structurally eliminated by using e_2 and e_4. The variables that are included in the resulting equation, let say e_6, is assumed to be all variables included in e_2 and e_4 except for the eliminated variable x_1. The equations e_2 and e_4 has been used and are therefore removed. The resulting structure can then be represented as

Equations	Original Equations	Unknown		
		x_1	x_2	x_3
e_1	$\{e_1\}$		X	
e_3	$\{e_3\}$		X	X
e_5	$\{e_5\}$			X
e_6	$\{e_2, e_4\}$		X	X

(9.49)

Next, the variable x_2 is structurally eliminated in all possible ways in (9.49). The variable x_2 is included in three equations, i.e., e_1, e_3, and e_6. The variable x_2 can be structurally eliminated by using any of the pairs of these three equation, i.e., $\{e_1, e_3\}$, $\{e_1, e_6\}$, and $\{e_3, e_6\}$. Then we get

Equations	Original Equations	Unknown		
		x_1	x_2	x_3
e_5	$\{e_5\}$			X
e_7	$\{e_1, e_3\}$			X
e_8	$\{e_1, e_2, e_4\}$			X
e_9	$\{e_2, e_3, e_4\}$			X

(9.50)

The variable x_1 can be structurally eliminated by using any of the 6 pairs of

Table 9.2: Matchings, redundant equations, and MSO sets computed by the second algorithm.

	Matching			RE	MSO Set
1	(x_1, e_2)	(x_2, e_1)	(x_3, e_3)	e_4	$\{e_1, e_2, e_3, e_4\}$
2				e_5	$\{e_1, e_3, e_5\}$
3			(x_3, e_4)	e_3	$\{e_1, e_2, e_3, e_4\}$
4				e_5	$\{e_1, e_2, e_4, e_5\}$
5			(x_3, e_5)	e_3	$\{e_1, e_2, e_3, e_5\}$
6				e_4	$\{e_1, e_2, e_4, e_5\}$
7		(x_2, e_3)	(x_3, e_4)	e_1	$\{e_1, e_2, e_3, e_4\}$
8				e_5	$\{e_2, e_3, e_4, e_5\}$
9			(x_3, e_5)	e_1	$\{e_1, e_3, e_5\}$
10				e_4	$\{e_2, e_3, e_4, e_5\}$
11	(x_1, e_4)	(x_2, e_1)	(x_3, e_3)	e_2	$\{e_1, e_2, e_3, e_4\}$
12				e_5	$\{e_1, e_2, e_4, e_5\}$
13			(x_3, e_5)	e_2	$\{e_1, e_2, e_4, e_5\}$
14				e_3	$\{e_1, e_3, e_5\}$
15		(x_2, e_2)	(x_3, e_3)	e_1	$\{e_1, e_2, e_3, e_4\}$
16				e_5	$\{e_2, e_3, e_4, e_5\}$
17			(x_3, e_5)	e_1	$\{e_1, e_2, e_4, e_5\}$
18				e_3	$\{e_2, e_3, e_4, e_5\}$
19		(x_2, e_3)	(x_3, e_5)	e_1	$\{e_1, e_3, e_5\}$
20				e_2	$\{e_2, e_3, e_4, e_5\}$

equations and we obtain:

Equations	Original Equations	Unknown		
		x_1	x_2	x_3
e_{10}	$\{e_1, e_3, e_5\}$			
e_{11}	$\{e_1, e_2, e_4, e_5\}$			
e_{12}	$\{e_2, e_3, e_4, e_5\}$			
e_{13}	$\{e_1, e_2, e_3, e_4\}$			
e_{14}	$\{e_1, e_2, e_3, e_4\}$			
e_{15}	$\{e_1, e_2, e_3, e_4\}$			

(9.51)

The two last equations are not stored, since their original equations sets are equal to the original equation set of e_{13}. Since no unknowns are included in the equations in (9.51) it follows that the corresponding sets of original equations are the MSO sets. It can be seen in (9.51) that this algorithm can find the same MSO set more than once.

Contrary to the new algorithm that uses a top-down approach, all these three algorithms use a bottom-up approach to find MSO sets. That is, the MSO sets are found by starting with a set containing one equation and then extend this set until an MSO set is found.

9.6.2 Computational Complexity Comparison

The first two algorithms are most similar and these two will next be compared. An example will illustrate a general way to determine that the computational complexity of the second algorithm is not better than for the first one. For further comparisons see (Rattfält, 2004). Therefore, the first algorithm can be considered when comparing the computational complexity of both the first and the second algorithm with the new algorithm developed in this thesis.

Example 9.8
As said before, all MSO sets are found with both these algorithms. However the first algorithm is computationally more efficient and this can be understood as follows. Consider row (a) in Table 9.1. This MSO set is constructed by the redundant equation e_1 and the matching shown there. The MSO set of row 15 in Table 9.2 is found by the same redundant equation and the same matching. For any row in Table 9.1, there exist at least one corresponding row in Table 9.2. All correspondences are:

- (a) to 15;
- (b) to 17;
- (c) to 7;
- (d) to 9 and 19;
- (e) to 20;
- (f) to 3, 5, 14, and 18;
- (g) to 1, 6, and 10; and
- (h) to 2, 4, 8, 12, and 16.

This is an example, but the method can be used for any example. Since each row in Table 9.1 corresponds to at least one row in Table 9.2 and are obtained by equivalent operations, it follows that the computational complexity of the second algorithm cannot be better than the computational complexity of the first algorithm.

9.6.3 Worst Case Analysis

In this section, the complexity of all the previous algorithms will be analyzed under the same condition as in the Theorem 9.10, i.e., for a fixed order of structural redundancy. The worst case, for all algorithms discussed in Section 9.6.1, is when all unknown variables are included in each equation.

In all the algorithms discussed in Section 9.6.1, a bottom-up approach is used and all subsets of MSO sets are traversed at least once in the worst case. For this case, the proper subsets of MSO sets are exactly those sets that are not PSO sets. The number of PSO sets grows polynomially in the number n of equations according to Lemma 9.6 and the estimate (9.47). Furthermore, the number of all subsets is 2^n. Hence for a fixed order of structural redundancy, the number of subsets of MSO sets grows exponentially and the computational complexity of these algorithms is exponential.

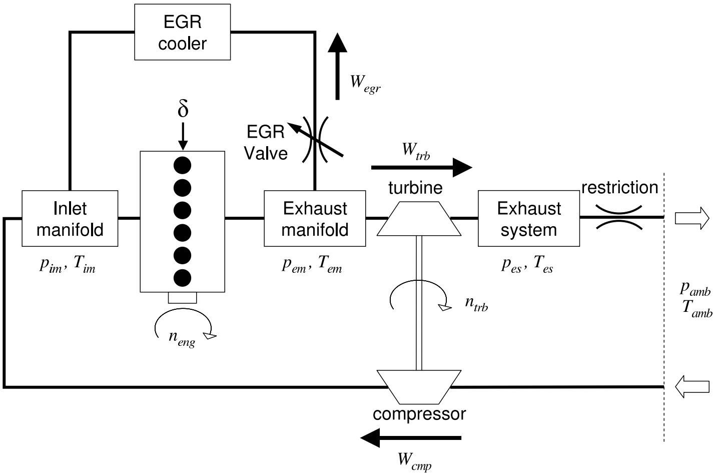

Figure 9.3: Example of a Scania truck engine.

For the algorithm presented in (Dustegör et al., 2006), it follows from the discussion in Section 9.6.1 that all maximal matchings have to be found. In the worst case, the number of maximal matchings is equal to the number of ordered subsets of equations with size $n - \varphi$, i.e., there are $n!/\varphi!$ number of maximal matchings. Hence, for a fixed order of structural redundancy, the computational complexity of this algorithm is factorial in the number of equations.

In conclusion, in the case of low structural redundancy, Algorithm 11 has better computational complexity than the others. However, it should be pointed out that, in the case of few unknowns, the roles are reversed. For a fixed number of unknowns, the new algorithm has exponential time complexity and all previous algorithms has polynomial time complexity. However, this situation is, as pointed out before, not common in industrial applications.

9.7 Application to a Large Industrial Example

To demonstrate the efficiency of the algorithm, described in the previous section, we will here apply it to an detail model of an industrial process. The process is a Scania truck diesel-engine and a sketch is shown in Figure 9.3. This engine has two actuators, namely the fuel injection δ and the EGR-valve. It has eight sensors, namely ambient pressure p_{amb}, ambient temperature T_{amb}, air flow W_{cmp}, inlet pressure p_{im}, inlet temperature T_{im}, exhaust pressure p_{em}, engine speed n_{eng}, and turbine speed n_{trb}. Further details of the application is presented in (Eriksson, 2004).

A simulation model of the engine was provided in Simulink. The model has 4 states and 4 outputs. These 4 outputs are W_{cmp}, p_{im}, p_{em}, and n_{trb}. The other 4 sensors are in the Simulink model implemented as inputs. To analyze the model, it was transferred to a flat list of equations. The number of equations is 126 and the structural redundancy is 4. The fact that the structural redundancy

Table 9.3: A comparison of three MSO algorithms.

Algorithm	Execution time
The old MSO algorithm	5900 s
The new basic algorithm	18 s
The new improved algorithm	0.42 s

is 4 is a consequence of that the number of outputs is 4.

For comparison, three algorithms were tested on the set of 126 equations. The first is the old MSO algorithm presented in (Krysander and Nyberg, 2002a), where an alternative partial reduction is used. Without any reduction, the old MSO algorithm is practically intractable for this example. The second is the new basic algorithm presented in Section 9.3 with the structural reduction in Section 9.4.1 applied initially, reducing the number of equations to 28. The third is the new improved algorithm presented in Section 9.4.

All algorithms were implemented in Matlab and executed on a PC with a 1 GHz processor. The execution times were measured in seconds and are presented in Table 9.3. There were 1419 MSO sets and in the table we can see that the new MSO algorithm is more than 14000 times faster than the old algorithm.

9.8 Conclusions

A new algorithm for finding all MSO sets of equations is developed. The proposed algorithm can use any structural representation presented in Section 8.5 for finding models with redundancy. There are three main ideas that are used in the new algorithm. First, it is based on a top-down approach as described in Section 9.3. Second, a structural reduction is used where subsets of equations are lumped together in order to reduce the size of the structural model. Third and last, it is prohibited that any MSO set is found more than once. For a fixed order of structural redundancy, the computational complexity of the new algorithm is polynomial in the number of equations, in contrast to previous algorithms where the complexity is at least exponential. The efficiency of the algorithm was demonstrated by applying the new and a previous algorithm to a model of a Scania truck engine.

10

S A MSO S D A E

One approach to design diagnosis systems is to use tests based on minimal rejectable models according to Chapter 2. When considering general non-linear dynamic models, it has been shown that structural methods can be used to identify MSO models with analytical redundancy. The most compact structural representation of dynamic models presented in Section 8.5 is not to distinguish different order of time-derivatives of the same variable as in (8.36). In this chapter we consider, given this type of structural representation, an MSO set of differential-algebraic equations. By differentiating equations, a new set of equations can be formed, such that the new set is an MSO set of algebraic equations with a structure where different order of derivatives are considered to be separate independent variables. This type of structure is a richer structure than the original structure and provides more information about dynamic models. An MSO set obtained in this way by differentiation of equations in an MSO set in the original representation will be called an *differentiated MSO set*.

There are two main reasons for computing differentiated MSO sets. First, a differential-algebraic model can be reduced to an algebraic model, and then also algebraic tools for test construction can be used. Second, high order derivatives of noisy signals are difficult to estimate accurately and look-up tables might be difficult to differentiate and these facts might imply that the consistency relation cannot be used in practice. Deriving consistency relations can be time-consuming or sometimes even not possible. Differentiated MSO sets can be used for predicting the highest order of derivatives in a consistency relation derived from the original MSO set. These prediction can then be used to select which MSO sets to derive consistency relations and test quantities from.

In this chapter, the existence of a differentiated MSO set is shown. It is desirable to differentiate each equation as few times as possible. We show that there exists a unique differentiated MSO set with the minimum highest order of derivatives for all equations. Since all highest order of derivatives of equations are minimized at the same time, we will call this set the *differentiated MSO set of minimum order*.

The outline of this chapter is as follows. First, Section 10.1 exemplifies a differentiated MSO set of minimum order. In Section 10.2, the reasons for computing differentiated MSO sets of minimum orders are further motivated. The structure of a differentiated MSO set of minimum order can be computed by utilizing only graph theoretical methods. A key concept in the graph theoretical approach is *structural differentiation* which is introduced in Section 10.3. This is then used to define an algorithm for computing the differentiated MSO set of minimum order in Section 10.4. Theory is then given in the remaining part of the chapter. Section 10.5 introduces basic definitions and notation. In Section 10.6, the existence of a differentiated MSO set is shown. In Section 10.7, the existence and the uniqueness of the differentiated MSO set of minimum order is proven. Finally, the conclusions are drawn in Section 10.8.

10.1 Differentiated MSO Set

First, we recall the two structural representations (8.36) and (8.38) and show an example of a differentiated MSO set. The following example will later also be used to define notation that will be needed.

Example 10.1
The differential-algebraic system

Equation	Expression		
e_1	$\lambda\, x + L\, m\, x^{(2)}$	=	0
e_2	$\lambda\, y + L\, m\, y^{(2)} + L\, m\, g$	=	0
e_3	$x^2 + y^2 - L^2$	=	0
e_4	$x\, y^{(1)} - y\, x^{(1)} - L^2\, z$	=	0

(10.1)

models the motion of a pendulum and an angular velocity measurement z. Here x, y, and λ are the unknown variables, L is the length, m is the mass, and g is the gravitational constant. The set of equations $\{e_1, e_2, e_3, e_4\}$ will be denoted E and the set of unknowns $\{x, y, \lambda\}$ by $\boldsymbol{X}$. The graph $G(E, \boldsymbol{X})$, for the system (10.1), represented as a biadjacency matrix is

Equation	Unknown $\boldsymbol{X}$			Known
E	x	y	λ	z
e_1	X		X	
e_2		X	X	
e_3	X	X		
e_4	X	X		X

(10.2)

By considering variables and their derivatives as separate independent variables, the biadjacency matrix of the bipartite graph for system (10.1) is

Equation	Unknown							Known
	x	$x^{(1)}$	$x^{(2)}$	y	$y^{(1)}$	$y^{(2)}$	λ	z
e_1	X		X				X	
e_2				X		X	X	
e_3	X			X				
e_4	X	X		X	X			X

(10.3)

Table 10.1: The differentiated MSO set of minimum order for the MSO set (10.1).

Equation		Unknown X_d										Known		
		X_d^l							X_d^m					
	E_d	x	$x^{(1)}$	$x^{(2)}$	y	$y^{(1)}$	$y^{(2)}$	λ	$x^{(3)}$	$y^{(3)}$	$\lambda^{(1)}$	z	$z^{(1)}$	$z^{(2)}$
E_d^l	e_1	X		X				X						
	e_2				X		X	X						
	e_3	X			X									
	$e_3^{(1)}$	X	X		X	X								
	$e_3^{(2)}$	X	X	X	X	X	X							
	e_4	X	X		X	X						X		
	$e_4^{(1)}$	X	X	X	X	X	X						X	
E_d^m	$e_1^{(1)}$	X	X					X	X		X			
	$e_2^{(1)}$				X	X		X		X	X			
	$e_3^{(3)}$	X	X	X	X	X	X		X	X				
	$e_4^{(2)}$	X	X	X	X	X	X		X	X				X

This algebraic system is not SO, but by differentiating equations with respect to t, new equations are obtained, for example

$$e_3^{(1)} : 2xx^{(1)} + 2yy^{(1)} = 0$$

The structure of the differentiated MSO set of minimum order is shown in Table 10.1. The three equation sets and the three variable sets in this table is not important now but will be explained later in Section 10.5.

In this particular case, the equations in the differentiated MSO set are all polynomials in the unknowns and the unknowns can therefore be eliminated using for example Gröbner basis (Cox et al., 1997). A consistency relation derived in this way is

$$m\, z^2(g^2 - L^2\,(z^{(1)})^2) - L^2\, m\,(z^{(2)})^2 = 0$$

It should be pointed out that the structural analysis outlined above is not restricted to polynomials and can be applied to general non-linear problems.

This example illustrates that by differentiating equations in an MSO set of differential-algebraic equation, a differentiated MSO set containing only algebraic equations can be obtained.

10.2 Use of Differentiated MSO Sets

There are several reasons for computing a differentiated MSO set and these reasons are discussed next.

10.2.1 Differentiated MSO Set for Test Construction

If a dynamic model contains non-polynomial differential-equations, polynomial differential-algebraic elimination methods cannot be used to obtain a consistency relation. However, by computing a differentiated MSO set, the task to derive a consistency relation is thereby reduced to an algebraic problem as seen in Example 10.1. Both Mathematica and Maple provide implementations for algebraic elimination handling also some non-polynomial equations.

Example 10.2
Consider the following MSO set of equations

Equation	Expression
e_1	$\dot{x}_1 = x_2^2 + u$
e_2	$x_2 = e^{x_1}$
e_3	$y = x_1$

(10.4)

where u and y are known, and x_1 and x_2 are unknown variables. Note that this model is non-polynomial since the second equations includes an exponential term. Neither of the two differential-algebraic implementations in Maple, i.e., `rifsimp` in the package `DEtools` and `Rosenfeld_Groebner` in the package `diffalg`, is able to eliminate x_1 and x_2 in this example.

The differentiated MSO set of minimum order is

Equation	Expression
e_1	$\dot{x}_1 = x_2^2 + u$
e_2	$x_2 = e^{x_1}$
e_3	$y = x_1$
$\dot{e}_3$	$\dot{y} = \dot{x}_1$

(10.5)

The unknowns x_1, $\dot{x}_1$, and x_2 can be eliminated algebraically in the equation system (10.5) by using for example the command `eliminate` in Maple. The consistency relation

$$\dot{y} - e^{2y} - u = 0 \tag{10.6}$$

is obtained. Hence by computing the differentiated MSO set of minimum order the elimination problem is reduced to an algebraic problem and algebraic elimination algorithms can also be used to obtain a consistency relation.

If elimination is undesirable or impossible, optimization methods can be used to check the consistency of the algebraic model based on the differentiated MSO set as the next example shows.

Example 10.3
Consider again the MSO set of equations (10.4) with the differentiated MSO set of minimum order (10.5). Let the four equations be described by $g_i = 0$ for $i \in \{1, 2, 3, 4\}$. Then a test quantity can be based on a least square estimate

$$T = \min_{[x_1, \dot{x}_1, x_2] \in \mathbb{R}^3} \sum_{i=1}^{4} g_i^2 \tag{10.7}$$

Numerical methods can be used for the minimization and there is no need to eliminate the unknowns. Note that it is a big advantage for the minimization that the relation between x_1 and $\dot{x}_1$ is algebraically described in the objective function.

10.2.2 Differentiated MSO Sets for Prediction

If the differential-algebraic equations are polynomials, then differential-algebraic elimination tools can be applied to the original MSO set to obtain a consistency relation. Even if it is theoretically possible to automatically compute a consistency relation, the following two problems can be encountered. First, differential-algebraic elimination algorithms require sometimes an unreasonable time to complete, or require more memory than is available (Wittkopf and Reid, 2001). Second, if a consistency relation has been derived, the consistency relation can include high order derivatives. High order derivatives of noisy signals are difficult to estimate accurately and this might imply that the consistency relation cannot be used in practice.

We have previously seen that the number of MSO sets can be large and that not all MSO sets need to be tested. Since, in general, not all MSO sets need to be tested and the elimination problem for the most complicated MSO sets might take an unreasonable long time to complete or result in a consistency relation with high order of derivatives, there is a need for a computational efficient method for predicting properties of the resulting consistency relation, without explicitly determining the consistency relation.

The structure of the original MSO set cannot be used to predict the order of derivatives of the known variables contained in a consistency relation derived from the MSO set. However the order of derivatives contained in the differentiated MSO set of minimum order is in the generic case equal to the order of derivatives contained in a consistency relation derived from the original MSO set. This will be illustrated in the next example.

Example 10.4
Consider the water-tank example with the model in Table 2.2. The structure is given in Table 8.2 and contains the feasible MSO sets given in Table 2.3. The differentiated MSO sets of minimum orders, obtained from the MSO set in the original structure, are shown in Table 10.2. The known variables included in each differentiated MSO set are shown in Table 10.3. By comparing the consistency relations in Table 2.4 with their corresponding differentiated MSO sets in Table 10.3, it can be concluded that the derivatives of the known variables included in the consistency relations are equal to the derivatives of the known variables contained in the differentiated MSO sets. Assume that second order derivatives can be estimated in the consistency relations, but not higher order derivatives. Table 10.3 then suggest that consistency relation based approach for test construction is not suitable for the 12:th MSO set.

Another property that can be investigated by the differentiated MSO set of minimum order is the fault influence of a test quantity based on the MSO set.

Example 10.5
Consider the third MSO set in Table 10.2. In the differentiated MSO set, the differentiated equation $e_6^{(1)}$ is included but not the original equation e_6. Assume that the water-level sensor has a constant bias fault. The influence of the fault can be expressed with a fault signal $f \neq 0$ such that

$$y_w = w + f \tag{10.8}$$

and $\dot{f} = 0$. From (10.8) and $\dot{f} = 0$, we get that $e_6^{(1)}$ is $\dot{y}_w = \dot{w}$. From this expression, it can be seen that the fault f will not influence the consistency of

Table 10.2: The MSO sets contained in (2.4) and their corresponding differentiated MSO sets of minimum orders.

i	MSO set	Differentiated MSO set of minimum order
1	$\{e_3, e_6, e_7\}$	$\{e_3, e_6, e_7\}$
2	$\{e_1, e_2, e_3, e_7\}$	$\{e_1, e_2, e_3^{(1)}, e_7, e_7^{(1)}\}$
3	$\{e_1, e_2, e_6, e_7\}$	$\{e_1, e_2, e_6^{(1)}, e_7\}$
4	$\{e_1, e_2, e_3, e_6\}$	$\{e_1, e_2, e_3, e_6, e_6^{(1)}\}$
5	$\{e_4, e_5, e_6, e_7\}$	$\{e_4, e_4^{(1)}, e_5, e_6, e_6^{(1)}, e_7, e_7^{(1)}\}$
6	$\{e_3, e_6, e_8, e_9\}$	$\{e_3, e_3^{(1)}, e_6, e_6^{(1)}, e_8^{(1)}, e_9\}$
7	$\{e_4, e_5, e_6, e_8, e_9\}$	$\{e_4, e_4^{(1)}, e_4^{(2)}, e_5, e_5^{(1)}, e_6, e_6^{(1)}, e_6^{(2)}, e_8^{(1)}, e_8^{(2)}, e_9, e_9^{(1)}\}$
8	$\{e_1, e_2, e_6, e_8, e_9\}$	$\{e_1^{(1)}, e_2^{(1)}, e_6^{(2)}, e_8^{(1)}, e_9\}$
9	$\{e_1, e_2, e_4, e_5, e_7\}$	$\{e_1, e_1^{(1)}, e_2, e_2^{(1)}, e_4^{(1)}, e_4^{(2)}, e_5, e_5^{(1)}, e_7, e_7^{(1)}, e_7^{(2)}\}$
10	$\{e_1, e_2, e_3, e_8, e_9\}$	$\{e_1, e_1^{(1)}, e_2, e_2^{(1)}, e_3^{(1)}, e_3^{(2)}, e_8^{(1)}, e_8^{(2)}, e_9, e_9^{(1)}\}$
11	$\{e_1, e_2, e_4, e_5, e_6\}$	$\{e_1, e_1^{(1)}, e_2, e_2^{(1)}, e_4, e_4^{(1)}, e_5, e_6, e_6^{(1)}, e_6^{(2)}\}$
12	$\{e_1, e_2, e_4, e_5, e_8, e_9\}$	$\{e_1, e_1^{(1)}, e_1^{(2)}, e_2, e_2^{(1)}, e_2^{(2)}, e_4^{(1)}, e_4^{(2)}, e_4^{(3)}, e_5, e_5^{(1)}, e_5^{(2)},$ $e_8^{(1)}, e_8^{(2)}, e_8^{(3)}, e_9, e_9^{(1)}, e_9^{(2)}\}$

$e_6^{(1)}$. This implies that the consistency of the differentiated MSO set 3 is not influenced by the fault either. Hence, no test based on this differentiated MSO set can detect a constant bias fault in e_6.

In general, if only differentiated versions of an original equation are included in the differentiated MSO of minimum order, then constant faults affecting the consistency of the equation will not be possible to detect.

10.3 Structural Differentiation

The structure in Table 10.1 represents a differentiated MSO set derived from the MSO set of equations (10.1). In Example 10.1 we showed that the structure in Table 10.1 could be obtained by differentiating equations. In this section, we show that the structure of a differentiated MSO set can be obtained without differentiating equations analytically, instead only structural operations will be used. The basic idea is to define a structural operation that, given the structure of an equation, is able to compute a correct structural representation of the differentiated equations. First, we compare the structural representations of an equation and its differentiated version in an example.

Example 10.6

Consider for example equation e_1 shown in (10.1) and equation $e_1^{(1)}$ with the analytical expression

$$L^{-1}(\lambda^{(1)}x + \lambda x^{(1)}) + mx^{(3)} = 0 \tag{10.9}$$

The variable $x^{(2)}$ is *linearly contained* in e_1 and $x^{(3)}$ is therefore linearly contained

Table 10.3: The MSO sets contained in (2.4) and known variables contained in their corresponding differentiated MSO sets of minimum orders.

i	MSO set	Known variables in the differentiated MSO
1	$\{e_3, e_6, e_7\}$	$\{y_w, y_q\}$
2	$\{e_1, e_2, e_3, e_7\}$	$\{u, y_q, y_q^{(1)}\}$
3	$\{e_1, e_2, e_6, e_7\}$	$\{u, y_q, y_w^{(1)}\}$
4	$\{e_1, e_2, e_3, e_6\}$	$\{u, y_w, y_w^{(1)}\}$
5	$\{e_4, e_5, e_6, e_7\}$	$\{y_w, y_w^{(1)}, y_q, y_q^{(1)}\}$
6	$\{e_3, e_6, e_8, e_9\}$	$\{y_w, y_w^{(1)}, y_q^{(1)}\}$
7	$\{e_4, e_5, e_6, e_8, e_9\}$	$\{y_w, y_w^{(1)}, y_w^{(2)}, y_q^{(1)}, y_q^{(2)}\}$
8	$\{e_1, e_2, e_6, e_8, e_9\}$	$\{u^{(1)}, y_w^{(2)}, y_q^{(1)}\}$
9	$\{e_1, e_2, e_4, e_5, e_7\}$	$\{u, u^{(1)}, y_q, y_q^{(1)}, y_q^{(2)}\}$
10	$\{e_1, e_2, e_3, e_8, e_9\}$	$\{u, u^{(1)}, y_q^{(1)}, y_q^{(2)}\}$
11	$\{e_1, e_2, e_4, e_5, e_6\}$	$\{u, u^{(1)}, y_w, y_w^{(1)}, y_w^{(2)}\}$
12	$\{e_1, e_2, e_4, e_5, e_8, e_9\}$	$\{u, u^{(1)}, u^{(2)}, y_q^{(1)}, y_q^{(2)}, y_q^{(3)}\}$

in equation $e_1^{(1)}$. Furthermore, both x and $x^{(1)}$ are nonlinearly contained in $e_1^{(1)}$ as a consequence of the fact that x is nonlinearly contained in e_1.

This example shows that variables are handled in different ways depending on if they are linearly or nonlinearly contained. To be able to take this different treatment into account, information about which variables that are linearly contained is added to the structural model. With this additional knowledge a *structural differentiation* can be defined that produce a correct structural representation of differentiated equations. Structural differentiation for an arbitrary variable x and an arbitrary equation e is defined in the following way:

a) If x is linearly contained in e then $x^{(1)}$ is linearly contained in $e^{(1)}$.

b) If x is nonlinearly contained in e then both x and $x^{(1)}$ are nonlinearly contained in $e^{(1)}$.

If x is linearly contained in e and $x^{(1)}$ is non-linearly contained in e, then $x^{(1)}$ is nonlinearly contained in $x^{(1)}$, i.e., rule (b) is dominant.

Example 10.7
Consider the pendulum in Example 10.1. The structural model (10.3) where information about which variables that are linearly contained is

Equation	Unknown							Known
	x	$x^{(1)}$	$x^{(2)}$	y	$y^{(1)}$	$y^{(2)}$	λ	z
e_1	n		l				n	
e_2				n		l	n	
e_3	n			n				
e_4	n	n		n	n			l

(10.10)

Table 10.4: The structure of the equations of the model in Table 2.2. Linearly included variables are denoted with l and non-linearly included variables with n.

Equation	Unknown								Known		
	q_1	w	$\dot{w}$	q_2	f_c	$\dot{f}_c$	f_{yf}	$\dot{f}_{yf}$	u	y_w	y_q
e_1	l								l		
e_2	l		l	l							
e_3		l		n							
e_4		l		n	n						
e_5						l					
e_6		l								l	
e_7				l							l
e_8				l			l				l
e_9								l			

Linearly included variables are denoted with l and non-linearly included variables with n. Structural differentiation of the first equation produces the structure

Equation	Unknown					
	x	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	λ	$\lambda^{(1)}$
e_1	n		l		n	
$e_1^{(1)}$	n	n		l	n	n

By comparing the structure of $e_1^{(1)}$ and the variables included in equation (10.9), we see that the structure is correct. By using structural differentiation multiple times, the structure in Table 10.1 is obtained.

The last example, in this section, considers the water-tank model.

Example 10.8
The structure shown in Table 10.4 has been used together with structural differentiation to compute the differentiated MSO sets of minimum order shown in Table 10.2.

10.4 Algorithm for Computing a Differentiated MSO Set

By starting with an MSO set E of equations and differentiating all equations with respect to t multiple times, we will later show that the set of differentiated equations will eventually grow faster than the set of differentiated variables. Therefore, it is always possible to obtain an SO set where different order of derivatives are distinguished and this set contains by definition an MSO subset.

An elementary algorithm to find an MSO set is to structurally differentiate all equations until there exists a subset that is an SO set. The Dulmage-Mendelsohn decomposition can be used to determine if there exists an SO subset. It follows from the results presented later in this chapter that this SO set is also an MSO set and that it is of minimum order.

In the algorithm we will use the notation

$$E^{(k)} = \{e^{(k)} | e \in E\} \tag{10.11}$$

The algorithm can then be summarized as follows.

Algorithm 12. `GetDifferentiatedMSO`(E)

```
M := E;
while $M^+ = \emptyset$ do
    $E := E^{(1)}$;
    $M := M \cup E$;
end while
return $M^+$
```

In the assignment $E := E^{(1)}$, it is implicitly assumed that the structure of the equations in $E^{(1)}$ are computed by using structural differentiation. The algorithm is motivated by the following theorem.

Theorem 10.1. *Algorithm 12 returns the differentiated MSO set of minimum order given an MSO set E of equations.*

The proof of this theorem is given in the end of Section 10.7. The computational complexity of computing a differentiated MSO is polynomial in the number of equations. All differentiated MSO sets of minimum orders have been computed by Algorithm 12 in this thesis.

10.5 Basic Definitions and Notation

Now, the theoretical part of this chapter starts. First some important structural properties will be defined.

Definition 10.1 (Structurally Singular Set)**.** *A finite set of equations E is* ***structurally singular (SS) set*** *with respect to the set of variables X if* $|E| > |\mathrm{var}_X(E)|$.

Definition 10.2 (Minimal Structurally Singular Set)**.** *A set of equations E is a* ***minimal structurally singular (MSS) set*** *with respect to X if E is structurally singular with respect to X and no proper subset of E is structurally singular with respect to X.*

Example 10.9

Equation system (10.1), with the structure (10.2), is an example of an MSS set with respect to $\boldsymbol{X} = \{x, y, \lambda\}$. This follows from the fact that $\{e_1, e_2, e_3, e_4\}$ is SS with respect to $\boldsymbol{X}$, but no proper subset is SS with respect to $\boldsymbol{X}$.

Before we present a characterization of MSS sets, a classical graph theoretical result is presented. The following theorem is often referred to as Hall's theorem (Harary, 1969).

Theorem 10.2 (System of Distinct Representatives)**.** *Let* $V = \{V_1, V_2, \cdots, V_m\}$ *be a set of objects and* $S = \{S_1, S_2, \cdots, S_n\}$ *a set of subsets of V. Then a complete matching of S into V exists if and only if* $\forall S' \subseteq S : |S'| \leq |\bigcup_{S_i \in S'} S_i|$.

We will use Theorem 10.2 in the following way.

Corollary 10.1. *There is a complete matching of E into X if and only if $|E'| \leq |\mathrm{var}_X(E')|$ for all $E' \subseteq E$.*

Proof. The corollary follows immediately from Theorem 10.2. □

The next theorem shows the relation between MSS sets and MSO sets.

Theorem 10.3. *A set M is an MSO set if and only if M is MSS with respect to $\boldsymbol{X}$ and $\mathrm{var}_{\boldsymbol{Z}}(M) \neq \emptyset$.*

Proof. Let M be an MSO set. First we will show that $G(M, \boldsymbol{X} \cup \boldsymbol{Z})$ has a complete matching of M into $\boldsymbol{X} \cup \boldsymbol{Z}$. Assume that $G(M, \boldsymbol{X} \cup \boldsymbol{Z})$ has not a complete matching of M into $\boldsymbol{X} \cup \boldsymbol{Z}$. Then there exists an $e \in M$ such that

$$\nu(G(M, \boldsymbol{X} \cup \boldsymbol{Z})) = \nu(G(M \setminus \{e\}, \boldsymbol{X} \cup \boldsymbol{Z}))$$

This and that $\nu(G(M, \boldsymbol{X})) \geq \nu(G(M \setminus \{e\}, \boldsymbol{X}))$ imply

$$\begin{aligned} \varphi_s(M \setminus \{e\}) &= \nu(G(M \setminus \{e\}, \boldsymbol{X} \cup \boldsymbol{Z})) - \nu(G(M \setminus \{e\}, \boldsymbol{X})) \geq \\ & \qquad \nu(G(M, \boldsymbol{X} \cup \boldsymbol{Z})) - \nu(G(M, \boldsymbol{X})) = \varphi_s(M) \end{aligned}$$

i.e., M is not a PSO set and therefore also not an MSO set. This is a contradiction and it follows that the graph $G(M, \boldsymbol{X} \cup \boldsymbol{Z})$ has a complete matching of M into $\boldsymbol{X} \cup \boldsymbol{Z}$.

This means that the structural redundancy of any subset E of M can be expressed as

$$\varphi_s(E) = |E| - \nu(G(E, \boldsymbol{X})) \tag{10.12}$$

i.e., equal to the expression in (9.5). Since the structural redundancy is expressed by (9.5) for any subset $E \subset M$, it follows from (9.14) that

$$\bar{\varphi}_s(E) \leq \varphi_s(E) \tag{10.13}$$

Since M is an MSO set, we get by definition that $\varphi_s(E) = 0$. This and (10.13) imply that $\bar{\varphi}_s(E) \leq 0$, i.e., E is not SS with respect to $\boldsymbol{X}$ according to Definition 10.1. Since $\bar{\varphi}_s(E) \leq 0$ for all $E \subset M$ and $\varphi_s(M) > 0$, it follows from (9.13) that $\bar{\varphi}_s(M) = \varphi_s(M) > 0$. This means according to Definition 10.1 that M is SS with respect to $\boldsymbol{X}$. From this and that each subset of M is not SS w.r.t. $\boldsymbol{X}$, we get from Definition 10.2 that M is MSS w.r.t. $\boldsymbol{X}$.

Since M is an MSO set, it follows that

$$\varphi_s(M) = \nu(G(M, \boldsymbol{X} \cup \boldsymbol{Z})) - \nu(G(M, \boldsymbol{X})) > 0 \tag{10.14}$$

This implies that $\mathrm{var}_{\boldsymbol{Z}}(M) \neq \emptyset$.

Now, we prove the opposite direction. Assume that M is MSS w.r.t. $\boldsymbol{X}$ and $\mathrm{var}_{\boldsymbol{Z}}(M) \neq \emptyset$. First we show that this implies that there exists a complete matching of M into $\boldsymbol{X} \cup \boldsymbol{Z}$ in $G(M, \boldsymbol{X} \cup \boldsymbol{Z})$. Let E be any proper subset of M. Any subset $E' \subseteq E \subset M$ is not SS w.r.t. $\boldsymbol{X}$, i.e., $|E'| \leq |\mathrm{var}_{\boldsymbol{X}}(E')|$. Since this is true for all subsets $E' \subseteq E$, it follows from Theorem 10.1 that there exists a complete matching of E into $\mathrm{var}_{\boldsymbol{X}}(E)$ in $G(E, \boldsymbol{X})$. Since E was an arbitrary proper subset of M, it follows that

$$\nu(G(E, \boldsymbol{X})) = |E| \tag{10.15}$$

for all $E \subset M$. Since $\text{var}_{\boldsymbol{Z}}(M) \neq \emptyset$, there exists an equation $e \in M$ such that $\text{var}_{\boldsymbol{Z}}(\{e\}) \neq \emptyset$. From (10.15) where E is substituted for $M \setminus \{e\}$, we get that

$$\nu(G(M \setminus \{e\}, \boldsymbol{X})) = |M - 1|$$

which together with $\text{var}_{\boldsymbol{Z}}(\{e\}) \neq \emptyset$ imply

$$\nu(G(M, \boldsymbol{X} \cup \boldsymbol{Z})) = |M| \tag{10.16}$$

This means that Assumption 9.1 holds for the set M and this implies that (9.13) is true for M and all its subsets.

Since M is MSS w.r.t. $\boldsymbol{X}$, it follows that

$$\bar{\varphi}_s(M) > 0 \tag{10.17}$$

and

$$\bar{\varphi}_s(E) \leq 0 \tag{10.18}$$

for all $E \subset M$. From (9.13) and (10.18), we get that $\varphi_s(E) = 0$ for all $E \subset M$. Furthermore, (9.13), (10.16), and (10.18) imply that $\varphi_s(M) = \bar{\varphi}_s(M) > 0$. Hence since $\varphi_s(E) = 0$ for all $E \subset M$ and $\varphi_s(M) = \bar{\varphi}_s(M) > 0$, it follows from Definition 8.5 that M is an MSO set. □

The definition of MSO sets is technically more involved than the definition of MSS sets. From now on, we will therefore use the notion of MSS set instead of the notion MSO set for simplifying the theoretical presentation. Concepts like, e.g., differentiated MSO set, will have a direct correspondence to differentiated MSS set and so on. The original set E of equations is always assumed to be an MSS set with respect to $\boldsymbol{X}$, as the one shown in (10.2).

To prove the existence of a differentiated MSS set the following notation is needed. Let $\boldsymbol{X}^{(k)}$ be the set of k:th order derivatives of the variables in $\boldsymbol{X}$, i.e.,

$$\boldsymbol{X}^{(k)} = \{x^{(k)} | x \in \boldsymbol{X}\} \tag{10.19}$$

Furthermore, let the set of all unknowns $\bar{\boldsymbol{X}}$, where different order of derivatives are distinguished, be defined as

$$\bar{\boldsymbol{X}} = \cup_{i=0}^{\infty} \boldsymbol{X}^{(i)} \tag{10.20}$$

Finally, given a set E of equations let a number ξ be defined by

$$\xi = \sum_{x \in \boldsymbol{X}} (1 + \max_{x^{(\alpha)} \in \text{var}_{\bar{\boldsymbol{X}}}(E)} \alpha) \tag{10.21}$$

This number is an upper limit of the number of variables in $\bar{\boldsymbol{X}}$ included in E. For the example (10.3), we get $\xi = 3 + 3 + 1 = 7$ which, in this case, is equal to the number of unknowns.

10.6 Existence of a Differentiated MSS Set

In this section we will show the existence of a differentiated MSS set. The intuition is as said before that by starting with an MSS set E w.r.t. $\boldsymbol{X}$ and differentiating all equations with respect to t multiple times, the set of differentiated equations will eventually grow faster than the set of differentiated variables. Therefore, it is always possible to obtain an SS set in this way and this set contains by definition an MSS subset.

An upper limit for the number of differentiations that are needed to obtain an MSS set is given in the next theorem.

Theorem 10.4. *Given an MSS set E with respect to $\boldsymbol{X}$, an upper limit for the number of differentiations that are needed to obtain an MSS set w.r.t. $\bar{\boldsymbol{X}}$ is given by*

$$m = 1 + \xi - |E| \tag{10.22}$$

Proof. First, we proof that m in (10.22) is an upper limit of the number of differentiations that are needed to obtain an SS set w.r.t. $\bar{\boldsymbol{X}}$.

If all equations are differentiated m times, the number of equations is

$$| \cup_{i=0}^{m} E^{(i)}| = (m+1)|E| \tag{10.23}$$

An upper limit for the number of unknowns $\bar{\boldsymbol{X}}$ in E is given by ξ, i.e.,

$$|\mathrm{var}_{\bar{\boldsymbol{X}}}(E)| \leq \xi \tag{10.24}$$

By differentiating all equations one time, we get a new derivative of highest order for each variable in $\boldsymbol{X}$. This and (10.24) imply that

$$|\mathrm{var}_{\bar{\boldsymbol{X}}}(\cup_{i=0}^{m} E^{(i)})| \leq \xi + m|\boldsymbol{X}| \tag{10.25}$$

By combining (10.23) and (10.25), we get that

$$| \cup_{i=0}^{m} E^{(i)}| - |\mathrm{var}_{\bar{\boldsymbol{X}}}(\cup_{i=0}^{m} E^{(i)})| \geq (m+1)|E| - \xi - m|\boldsymbol{X}| = (|E| - \xi) + m(|E| - |\boldsymbol{X}|) \tag{10.26}$$

Since E is MSS w.r.t. $\boldsymbol{X}$, we have that $|E| = |\boldsymbol{X}|+1$. This, (10.26), and Definition 10.1 imply that $\cup_{i=0}^{m} E^{(i)}$ is an SS set if

$$|E| - \xi + m > 0 \tag{10.27}$$

The minimum integer m that fulfills (10.27) is given by (10.22). Hence, m in (10.22) is an upper limit of the number of differentiations that are needed to obtain an SS set w.r.t. $\bar{\boldsymbol{X}} = \cup_{i=0}^{\infty} \boldsymbol{X}^{(i)}$. By Definition 10.2, it then follows that there exists an MSS subset of $\cup_{i=0}^{m} E^{(i)}$ w.r.t. $\bar{\boldsymbol{X}}$ and this completes the proof. □

In the case of an MSS set of static equations, we have that $\xi = |\boldsymbol{X}|$ and $|E| = |\boldsymbol{X}| + 1$. Then we get $m = 1 + \xi - |E| = 0$ as expected.

The existence of an upper limit of the number of differentiations that are needed to obtain a differentiated MSS set is sufficient for the existence of a differentiated MSS set as the next corollary shows.

Corollary 10.2. *Given an MSS set E w.r.t. $\boldsymbol{X}$, there exists an MSS set $E_d \subseteq \cup_{i=0}^{m} E^{(i)}$ with respect to $X_d = \mathrm{var}_{\bar{\boldsymbol{X}}}(E_d)$ where m is given by (10.22).*

Proof. The corollary follows directly from Theorem 10.4. □

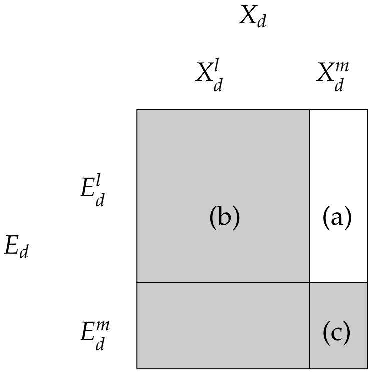

Figure 10.1: Partition of the graph.

Corollary 10.2 implies that Algorithm 12 will terminate in at most m iterations.

Example 10.10
For the example (10.3), $|E| = 4$, $|\mathbf{X}| = 3$, and $\xi = 7$. The upper limit m in (10.22) is $m = 4$. After differentiating all four equations four times, 20 equations are obtained with 19 unknowns. Hence, this set contains an MSS set. However, all these 20 equations are not needed to find a differentiated MSS set. Table 10.1 shows an MSS set with only 11 equations and 10 unknowns where the order of all the derivatives are at most three.

This example shows that there might exist differentiated MSS sets where the orders of all the derivatives are strictly less than the upper limit given in Theorem 10.4. A natural question is then: Does there exist a differentiated MSS set where the highest order of derivatives of all equations are minimized, or given a differentiated MSS set is it possible to decrease the highest order of derivative of one equation by increasing the highest order of derivative of another equation to obtain another differentiated MSS set? These questions will be answered in the following section.

10.7 Uniqueness of Differentiated MSS Sets

In the previous section, the existence of a differentiated MSS set E_d w.r.t X_d was shown. In the following section some aspects of uniqueness are investigated.

From now on, we assume that the equations E have been differentiated and an MSS set E_d with respect to X_d has been found as in Table 10.1. The set E_d is partitioned into two sets E_d^l and E_d^m, where E_d^m contains the highest derivative of each equation in E. The set X_d is partitioned into X_d^l and X_d^m in a similar way.

In general, the bipartite graph can be partitioned as in Figure 10.1, using the introduced notation. The structure of the sub-graphs (a), (b), and (c) are revealed in a sequence of lemmas, which leads to the main result formulated in Theorem 10.5. There, it is shown that there exists a unique differentiated MSS set of minimum order. The set is of minimum order in the following sense. For any other differentiated MSS set, derived from the same original set of equations, the order of the highest derivative of each equation is strictly greater than the order of the derivatives of the same equation in the MSS set of minimum order.

Now, the sequence of lemmas, mentioned above, will be presented. The first result is that there is only one redundant equation in an MSS set.

Lemma 10.1. *If E is an MSS set w.r.t. X, then* $|E| = |\mathrm{var}_X(E)| + 1$.

Proof. Since E is SS w.r.t. X, it follows that

$$|E| \geq |\mathrm{var}_X(E)| + 1$$

If equality holds, then there is nothing to prove. Assume that E is MSS w.r.t. X and that

$$|E| > |\mathrm{var}_X(E)| + 1$$

Take any $E' \subset E$ such that

$$|E'| = |\mathrm{var}_X(E)| + 1$$

Since $E' \subset E$, it follows that $|\mathrm{var}_X(E')| \leq |\mathrm{var}_X(E)|$ which implies that

$$|E'| = |\mathrm{var}_X(E)| + 1 \geq |\mathrm{var}_X(E')| + 1$$

This means that E' is SS which contradicts the assumption and the lemma follows. □

The next Lemma shows that the sub-graph (a) in Figure 10.1 has no edges.

Lemma 10.2. $\mathrm{var}_{X_d^m}(E_d^l) = \emptyset$

Proof. Assume that

$$x_i^{(l)} \in \mathrm{var}_{X_d^m}(E_d^l) \subset X_d^m$$

Then $x_i^{(l)} \in \mathrm{var}_{X_d^m}(e_j^{(k)})$ for some $e_j^{(k)} \in E_d^l$. Since $e_j^{(k)} \in E_d^l$, it follows that $e_j^{(k+p)} \in E_d^m$ for some $p \in \mathbb{Z}_+$. This implies that $x_i^{(l+p)} \in \mathrm{var}_{X_d}(E_d)$. But this contradicts the assumption that $x_i^{(l)} \in X_d^m$, which completes the proof. □

Now we show that there is one more equation than unknowns in X_d^m in the sub-graph (c). It is also shown that the degree of the variable nodes in (c) are nonzero.

Lemma 10.3. $|E_d^m| = |\mathrm{var}_{X_d^m}(E_d^m)| + 1$ *and* $\mathrm{var}_{X_d^m}(E_d^m) = X_d^m$.

Proof. From the definition of X_d^m it follows that

$$|\boldsymbol{X}| = |X_d^m| \tag{10.28}$$

and $X_d^m = \mathrm{var}_{X_d^m}(E_d)$. This,

$$\mathrm{var}_{X_d^m}(E_d) = \mathrm{var}_{X_d^m}(E_d^l) \cup \mathrm{var}_{X_d^m}(E_d^m)$$

and Lemma 10.2 imply

$$X_d^m = \mathrm{var}_{X_d^m}(E_d^m) \tag{10.29}$$

which is the second conclusion of this lemma.

From the definition of E_d^m, it follows that

$$|E_d^m| = |E| \tag{10.30}$$

The definition of $\boldsymbol{X}$, (10.28), and (10.29) imply

$$|\mathrm{var}_{\boldsymbol{X}}(E)| = |\mathrm{var}_{X_d^m}(E_d^m)| \tag{10.31}$$

Since E is MSS w.r.t. $\boldsymbol{X}$, Lemma 10.1 implies

$$|E| = |\mathrm{var}_{\boldsymbol{X}}(E)| + 1 \tag{10.32}$$

Now, eliminating $|E|$ and $|\mathrm{var}_{\boldsymbol{X}}(E)|$ by using (10.30) and (10.31),

$$|E_d^m| = |\mathrm{var}_{X_d^m}(E_d^m)| + 1$$

is obtained and the Lemma follows. □

The next lemma states that the cardinality of the two node sets in the subgraph (b) are the same and that the degree of the variable nodes in (b) are nonzero. That the degrees of the equation nodes are nonzero follows trivially from Lemma 10.2 and the fact that each equation has to contain at least one unknown. Otherwise the equation itself would be an MSS sets w.r.t. X_d.

Lemma 10.4. $|E_d^l| = |\mathrm{var}_{X_d}(E_d^l)|$ *and* $\mathrm{var}_{X_d}(E_d^l) = X_d^l$.

Proof. Lemma 10.1 applied to the MSS set E_d implies that

$$|E_d^m| + |E_d^l| = |X_d^m| + |X_d^l| + 1$$

and Lemma 10.3 implies that

$$|E_d^m| = |X_d^m| + 1$$

From these two equalities, it follows that

$$|E_d^l| = |X_d^l| \tag{10.33}$$

Since E_d is an MSS set with respect to X_d, it follows that $E_d^l \subsetneq E_d$ is not SS with respect to X_d, i.e.,

$$|E_d^l| \leq |\mathrm{var}_{X_d}(E_d^l)| \tag{10.34}$$

Lemma 10.2 implies that

$$\mathrm{var}_{X_d}(E_d^l) = \mathrm{var}_{X_d^l}(E_d^l) \subset X_d^l$$

By using this in (10.34), it follows that

$$|E_d^l| \leq |\mathrm{var}_{X_d}(E_d^l)| \leq |X_d^l| \tag{10.35}$$

This and (10.33) imply that

$$|E_d^l| = |\mathrm{var}_{X_d}(E_d^l)| = |X_d^l|$$

Finally, since $\mathrm{var}_{X_d}(E_d^l) \subset X_d^l$ and $|\mathrm{var}_{X_d}(E_d^l)| = |X_d^l|$ the lemma follows. □

In the example in Table 10.1, the sub-graph (c) is isomorphic to the original graph in (10.2). In general, the edges of (c) is a subset of the set of edges corresponding to the original graph. However, the following result shows that (c) still represents an MSS set.

Lemma 10.5. *The set E_d^m is an MSS set w.r.t. X_d^m.*

Proof. Assume that E_{d1}^m is SS w.r.t. X_d^m and $E_{d1}^m \subsetneq E_d^m$. The idea is to show that these assumptions imply that $E_{d1}^m \cup E_d^l$ is SS w.r.t. X_d which contradicts that E_d is MSS w.r.t. X_d. The assumption that E_{d1}^m is SS w.r.t. X_d^m and Lemma 10.4 imply that

$$\begin{aligned} |E_{d1}^m \cup E_d^l| &< |\mathrm{var}_{X_d^m}(E_{d1}^m)| + |\mathrm{var}_{X_d^l}(E_d^l)| \\ &= |\mathrm{var}_{X_d^m}(E_{d1}^m) \cup \mathrm{var}_{X_d^l}(E_d^l)| \end{aligned} \tag{10.36}$$

From Lemma 10.4, it follows that

$$\mathrm{var}_{X_d^l}(E_{d1}^m) \subset \mathrm{var}_{X_d^l}(E_d^l) = X_d^l$$

From this, Lemma 10.2, and that

$$\mathrm{var}_{X_d^l}(E_{di}) \cup \mathrm{var}_{X_d^m}(E_{di}) = \mathrm{var}_{X_d}(E_{di})$$

for any E_{di}, it follows that

$$\mathrm{var}_{X_d^m}(E_{d1}^m) \cup \mathrm{var}_{X_d^l}(E_d^l) = \mathrm{var}_{X_d}(E_{d1}^m \cup E_d^l)$$

If the left-hand side of this expression is substituted into (10.36), then it follows that $E_{d1}^m \cup E_d^l$ is SS w.r.t. X_d which contradicts that E_d is an MSS set w.r.t. X_d. Hence the lemma follows. □

Consider two different MSS sets derived from the same equations. It follows from the next result that the two sub-graphs (c) in Figure 10.1 corresponding to the two MSS sets are isomorphic.

Lemma 10.6. *There exist integers $\alpha_1, \ldots, \alpha_n$ such that for any MSS set E_d w.r.t. $\bar{X}$ derived from $E = \{e_1, \ldots, e_n\}$, the set E_d^m is*

$$E_d^m = \{e_1^{(\alpha_1+k)}, \ldots, e_n^{(\alpha_n+k)}\}$$

for some integer k.

Proof. Let E_{d1} and E_{d2} be two arbitrary MSS sets w.r.t. $\bar{X}$ with the corresponding subsets

$$E_{d1}^m = \{e_1^{(\alpha_1)}, \ldots, e_n^{(\alpha_n)}\}$$

and

$$E_{d2}^m = \{e_1^{(\beta_1)}, \ldots, e_n^{(\beta_n)}\}$$

To prove the lemma, it is sufficient to show that $\beta_i - \alpha_i = k$ for some k. Let $k = \max_i(\beta_i - \alpha_i)$. Either $\alpha_i = \beta_i$ for all i and there is nothing to prove, or the MSS sets can be enumerated so that $k > 0$. We can therefore assume that $k > 0$. Let

$$E_0 = \{e_i^{(\alpha_i)} : \beta_i - \alpha_i = k\}$$

and

$$X_0 = \mathrm{var}_{X_{d1}^m}(E_0)$$

It holds that

$$E_0^{(k)} := \{e_i^{(\alpha_i+k)} : \beta_i - \alpha_i = k\} = \{e_i^{(\beta_i)} : \beta_i - \alpha_i = k\}$$

and consequently it follows that

$$E_0^{(k)} \subset E_{d2}^m \tag{10.37}$$

Recall that $\beta_i = \alpha_i + k$ for $e^{(\beta_i)} \in E_0^{(k)}$ and that
$\beta_i < \alpha_i + k$ for $e^{(\beta_i)} \in E_{d2}^m \backslash E_0^{(k)}$. Assume that

$$x_i^{(\gamma)} \in X_0 = \mathrm{var}_{X_{d1}^m}(E_0)$$

It follows that $x_i^{(\gamma+k)} \in X_{d2}^m$ and hence

$$X_0^{(k)} \subset X_{d2}^m \tag{10.38}$$

It follows also that $x_i^{(\gamma+k)} \notin \mathrm{var}_{X_{d2}^m}(E_{d2}^m \backslash E_0^{(k)})$ and consequently it holds that

$$\mathrm{var}_{X_0^{(k)}}(E_{d2}^m \setminus E_0^{(k)}) = \emptyset \tag{10.39}$$

Assume now that $\alpha_i - \beta_i = k$ does not hold for all i or equivalently

$$E_0 \neq E_{d1}^m \tag{10.40}$$

We will show that this contradicts (10.39). The set E_{d1}^m is an MSS set w.r.t. X_{d1}^m according to Lemma 10.5. Together with assumption (10.40) this implies that $|E_0| \leq |X_0|$ and

$$|E_0^{(k)}| \leq |X_0^{(k)}|$$

Moreover E_{d2}^m is an MSS set w.r.t. X_{d2}^m according to Lemma 10.5 and hence

$$|E_{d2}^m| > |X_{d2}^m|$$

It follows from the two inequalities above and the set relations (10.37) and (10.38) that

$$|E_{d2}^m \setminus E_0^{(k)}| = |E_{d2}^m| - |E_0^{(k)}| > |X_{d2}^m| - |X_0^{(k)}| = |X_{d2}^m \setminus X_0^{(k)}|$$

This implies that

$$|E_{d2}^m \setminus E_0^{(k)}| > |\mathrm{var}_{X_{d2}^m \backslash X_0^{(k)}}(E_{d2}^m \setminus E_0^{(k)})|$$

and since $E_{d2}^m \setminus E_0^{(k)}$ is not SS we have

$$|E_{d2}^m \setminus E_0^{(k)}| \leq |\mathrm{var}_{X_{d2}^m}(E_{d2}^m \setminus E_0^{(k)})|$$

It follows from these two inequalities that

$$\begin{aligned}
|\mathrm{var}_{X_0^{(k)}}(E_{d2}^m \setminus E_0^{(k)})| =& |\mathrm{var}_{X_{d2}^m}(E_{d2}^m \setminus E_0^{(k)})| \\
& - |\mathrm{var}_{X_{d2}^m \backslash X_0^{(k)}}(E_{d2}^m \setminus E_0^{(k)})| \\
>& |E_{d2}^m \setminus E_0^{(k)}| - |E_{d2}^m \setminus E_0^{(k)}| = 0
\end{aligned}$$

This contradicts (10.39) and the proof is complete. □

Table 10.5: All differentiated MSS sets up to order two of the MSS set $\{e_3, e_6, e_8, e_9\}$ in (2.4).

MSO set
$\{e_3, e_3^{(1)}, e_6, e_6^{(1)}, e_8^{(1)}, e_9\}$
$\{e_3^{(1)}, e_3^{(2)}, e_6^{(1)}, e_6^{(2)}, e_8^{(1)}, e_8^{(2)}, e_9, e_9^{(1)}\}$
$\{e_3, e_3^{(2)}, e_6, e_6^{(2)}, e_8^{(1)}, e_8^{(2)}, e_9, e_9^{(1)}\}$
$\{e_3, e_3^{(1)}, e_3^{(2)}, e_6, e_6^{(1)}, e_6^{(2)}, e_8^{(2)}, e_9^{(1)}\}$

Next an example will be used to illustrate the consequence of Lemma 10.6.

Example 10.11
Consider the MSS set $\{e_3, e_6, e_8, e_9\}$ in (2.4). Table 10.5 shows all differentiated MSS sets included in the system obtained when differentiating all equations two times. The first differentiated MSS set is the differentiated MSS set of minimum order. In this set, the set of highest order of derivatives is

$$\{e_3^{(1)}, e_6^{(1)}, e_8^{(1)}, e_9\} \tag{10.41}$$

The other MSS sets have the same set

$$\{e_3^{(2)}, e_6^{(2)}, e_8^{(2)}, e_9^{(1)}\} \tag{10.42}$$

of highest order of derivatives. The set (10.42) can be obtained by differentiating all equations in (10.41) exactly one time. Lemma 10.6 states that for any differentiated MSS set the maximum order of derivatives are given by

$$\{e_3^{(1+k)}, e_6^{(1+k)}, e_8^{(1+k)}, e_9^{(k)}\} \tag{10.43}$$

for some integer k.

This example shows that given a differentiated MSO set, it is not possible to decrease the highest order of derivative of one equation by increasing the highest order of derivative of another equation to obtain another differentiated MSO set. Note also that, there exists only one MSS sets where the maximum order of derivatives can be expressed with the minimum number $k = 0$ in (10.43). This is generally true according to the next theorem.

Theorem 10.5. *Given an MSS set E w.r.t. $\mathbf{X}$, there exists a unique differentiated MSS set E_d w.r.t. $\bar{\mathbf{X}}$ of minimum order.*

Proof. Assume that E_{d1} and E_{d2} are two differentiated MSS sets of minimum order. According to Lemma 10.6 the corresponding sets E_{d1}^m and E_{d2}^m coincide and the notation E_d^m is used for both. Let X_d' be defined as $X_{d1} \cup X_{d2}$. The set $E_{d1}^l \cup E_{d2}^l$ is not SS, since this would imply that there exists a subset of $E_{d1}^l \cup E_{d2}^l$ that is an MSS set, which contradicts that E_{d1} and E_{d2} are both of minimum order. Hence

$$|E_{d1}^l \cup E_{d2}^l| \leq |\mathrm{var}_{X_d'}(E_{d1}^l \cup E_{d2}^l)|$$

Using this inequality, Lemma 10.4, and that

$$\mathrm{var}_{X'_d}(E^l_{d1} \cup E^l_{d2}) = \mathrm{var}_{X'_d}(E^l_{d1}) \cup \mathrm{var}_{X'_d}(E^l_{d2})$$

we get

$$\begin{aligned} |E^l_{d1} \cap E^l_{d2}| &= |E^l_{d1}| + |E^l_{d2}| - |E^l_{d1} \cup E^l_{d2}| \\ &\geq |\mathrm{var}_{X'_d}(E^l_{d1})| + |\mathrm{var}_{X'_d}(E^l_{d2})| \\ &\quad -|\mathrm{var}_{X'_d}(E^l_{d1}) \cup \mathrm{var}_{X'_d}(E^l_{d2})| \\ &= |\mathrm{var}_{X'_d}(E^l_{d1}) \cap \mathrm{var}_{X'_d}(E^l_{d2})| \end{aligned} \tag{10.44}$$

The set relation

$$\mathrm{var}_{X'_d}(E_{d1} \cap E_{d2}) \subset \mathrm{var}_{X'_d}(E_{d1}) \cap \mathrm{var}_{X'_d}(E_{d2})$$

holds and it follows from Lemma 10.4 that

$$\mathrm{var}_{X'_d}(E_{d1}) \cap \mathrm{var}_{X'_d}(E_{d2}) = (\mathrm{var}_{X'_d}(E^l_{d1}) \cap \mathrm{var}_{X'_d}(E^l_{d2})) \cup \mathrm{var}_{X^m_d}(E^m_d)$$

where $\mathrm{var}_{X'_d}(E^l_{d1}) \cap \mathrm{var}_{X'_d}(E^l_{d2})$ and $\mathrm{var}_{X^m_d}(E^m_d)$ are disjoint according to Lemma 10.2. This gives that

$$|\mathrm{var}_{X'_d}(E_{d1} \cap E_{d2})| \leq |\mathrm{var}_{X'_d}(E_{d1}) \cap \mathrm{var}_{X'_d}(E_{d2})| = |\mathrm{var}_{X'_d}(E^l_{d1}) \cap \mathrm{var}_{X'_d}(E^l_{d2})| + |\mathrm{var}_{X^m_d}(E^m_d)|$$

where

$$|\mathrm{var}_{X'_d}(E^l_{d1}) \cap \mathrm{var}_{X'_d}(E^l_{d2})| \leq |E^l_{d1} \cap E^l_{d2}|$$

according to (10.44) and

$$|\mathrm{var}_{X^m_d}(E^m_d)| < |E^m_d|$$

according to Lemma 10.3. It follows that

$$\begin{aligned} |\mathrm{var}_{X'_d}(E_{d1} \cap E_{d2})| &< |E^l_{d1} \cap E^l_{d2}| + |E^m_d| \\ &= |(E^l_{d1} \cap E^l_{d2}) \cup E^m_d| \\ &= |E_{d1} \cap E_{d2}| \end{aligned}$$

Hence, $E_{d1} \cap E_{d2}$ is an SS set and can not be a proper subset of the MSS sets E_{d1} and E_{d2}. It follows that

$$E_{d1} = E_{d2} = E_{d1} \cap E_{d2}$$

and the proof is complete. □

Algorithm 12 differentiates all equations until there exists a subset that is an SO set. Theorem 10.5 implies that this SO sets is the differentiated MSO set of minimum order.

Now, we are ready to prove Theorem 10.1 which states that the output set M^+ of Algorithm 12 is the differentiated MSO set of minimum order.

Proof. Given an MSO set E, it follows that E is MSS with respect to $\boldsymbol{X}$ and that $\mathrm{var}_Z(E) \neq \emptyset$ according to Theorem 10.3. From Theorem 10.5, it follows that there exists a unique differentiated MSS sets E_d with respect to $\bar{\boldsymbol{X}}$ of minimum order. Lemma 10.3 implies that some derivative of each equation in E is contained in

E_d. From this and the fact that $\mathrm{var}_Z(E) \neq \emptyset$, it follows that $\mathrm{var}_Z(E_d) \neq \emptyset$. Hence E_d is the unique differentiated MSO set of minimum order. From Theorem 9.6, it follows that $E_d \subseteq M^+$ if $E_d \subseteq M$. Since there exists a unique differentiated MSS set w.r.t. $\bar{X}$ of minimum order according to Theorem 10.5, E_d is the only MSO set contained in M^+. The set M^+ is a PSO set according to Lemma 9.1. The structural redundancy $\varphi_s(M^+)$ is one because otherwise, there would be more than one MO subsets of M^+ contradicting the uniqueness of E_d. Hence, it follows that $M^+ = E_d$. □

10.8 Conclusions

One approach for design of diagnosis systems is to use tests based on minimal rejectable models. When considering general non-linear dynamic models, structural methods can be used to identify MSO models with analytical redundancy.

A method has been presented that transforms an MSO set of differential-algebraic equations into an MSO set of equations where all the unknowns are algebraic. This is done by considering the unknowns and their derivatives as separate independent variables and then differentiating equations to obtain a differentiated MSO set.

To present the structure of the algebraic system, a bipartite graph is used and properties of the graph have been investigated in a sequence of lemmas. It is desirable to differentiate the equations as few times as possible, to avoid higher derivatives of measured signals. The main result is stated in Theorem 10.5, where it is shown that there exists a unique differentiated MSO set of minimum order. Given an MSO set, Algorithm 12 returns the unique differentiated MSO set of minimum order. This algorithm is purely structural and is based on the introduced concept of structural differentiation.

It has been illustrated that differentiated MSO sets of minimum order can be used for two main reasons. Deriving test quantities can be time-consuming and difficult. Differentiated MSO sets of minimum order is easily computed and can be used to suggest which MSO sets to derive consistency relations from. Furthermore, the differentiated MSO sets can be used to derive consistency relations, by using algebraic elimination methods or by using optimization methods.

11

T S B I A D S

Fault isolability analysis refers to the question of which faults that can be distinguished from other faults. This information is important because if the correct fault can be identified then correct accommodation or repair actions can be taken for that fault. Thereby, for example the impact of severe faults might be reduced or repair-time can be decreased.

When designing diagnosis systems by starting from the model as described in Section 2.3.3, a set of rejectable models is found. In previous chapters, we have seen that the number of rejectable models can be large and that it is sufficient to test a subset of those to obtain soundness. Soundness has been the criteria for selecting models to test and this is a strong requirement that for some applications might be unrealistic to obtain. In this chapter, we consider a set of potential tests, for example one for each rejectable model, and one goal is to select a subset of those potential tests with maximum fault isolability. This is, as we will see later, a weaker requirement than to obtain soundness.

The maximum fault isolability is equal to the isolability obtained by using all potential tests. This isolability can be computed by using test quantities, rejection regions, and test assumptions for all potential tests. However, typically a potential test corresponds to a rejectable model M_i, for which the test assumption $\Phi_i = \text{assump}(M_i)$ is easily computed, but a test quantity T_i and a rejection region $\mathcal{R}_i$ might be difficult to derive. In this chapter, a method for computing a prediction of the isolation capability of a diagnosis system is proposed. This prediction is only based on the test assumption Φ_i which means that the test quantity and the rejection region are not needed in the prediction. Therefore, this approximation can serve as guidance for selecting tests such that good fault isolation of the resulting diagnosis system is obtained. Furthermore, this chapter includes a method to compute the exact fault isolability for a diagnosis system given its fault influence of each test in the diagnosis system. If for example some selected tests have been derived, this method can be used to compute the actual fault isolability obtained.

The outline of this chapter is as follows. In Section 11.1, we formally define the analytical isolability of a diagnosis system. Section 11.2 introduces desired isolability that is a useful and intuitive way to express design specifications of a diagnosis system. Then, by using the desired isolability, problems to be solved in this chapter are posed, for example how to select tests such that the desired isolability can be obtained. In Section 11.3 an isolability prediction called structural isolability is introduced. It is proved that the structural isolability is a necessary condition for analytical isolability. Thereby it is possible to compute the structural isolability to get an upper bound for the analytical isolability. In Section 11.4, the structural isolability is written on a form suitable for computation and Section 11.5 describes different properties and representations of isolability. Section 11.6 reveals basic properties about the analytical isolability before the analytical and the structural isolability are compared in Section 11.7. There, it is shown that the structural and analytical can be different and in Section 11.8, a sufficient condition for equality is given. In Section 11.9, we answer the questions posed in Section 11.2 and finally the conclusions are drawn.

11.1 Analytical Isolation Capability

In this section, we will formally define and illustrate definitions concerning the fault isolation capability of a diagnosis system. We start to define two such properties of a diagnosis systems.

Definition 11.1 (Isolable with a Diagnosis System). *A system behavioral-mode b_i is **isolable** from another system behavioral-mode b_j with a diagnosis system, if there exists some observation $z \in \mathbb{Z}$ such that b_i is a candidate but b_j is not.*

Definition 11.2 (Detectable with a Diagnosis System). *A fault mode b_i is **detectable** with a diagnosis system if b_i is isolable from the no-fault system behavioral mode with the diagnosis system.*

When there is no risk of confusion, we say for example that a fault is detectable without explicitly pointing out which diagnosis system that is used. To summarize which behavioral modes that are isolable from others and which faults that are detectable with a diagnosis system, the following binary relation will be used.

Definition 11.3 (Analytical Isolability of a Diagnosis System). *Let Δ be a diagnosis system. A binary relation $I(\Delta)$ on $\mathbf{B}\times\mathbf{B}$ is the **analytical isolability of the diagnosis system** Δ if*

$$I(\Delta) = \{(b_1, b_2) | \exists z \in \mathbb{Z} : \big(b_1 \in C(z) \wedge b_2 \notin C(z)\big)\} \tag{11.1}$$

The interpretation of $(b_1, b_2) \in I(\Delta)$ is that b_1 is isolable from b_2 with the diagnosis system Δ. If $(b, \mathbf{NF}) \in I(\Delta)$ for a fault mode b, then it means that b is detectable with the diagnosis system Δ. Next an example illustrates the definition of analytical isolability.

Example 11.1
Consider the following diagnosis model:

Assumption	Equation	Expression
Sensor 1		
$S_1 = \text{NF}$	e_1	$z_1 = x_1$
Component		
	e_2	$x_1 = x_2^2$
Sensor 2		
$S_2 = \text{NF}$	e_3	$z_2 = x_2$

(11.2)

with the possible component behavioral modes defined by:

Component	Possible behavioral modes
Sensor 1	$S_1 \in \{\text{NF}, \text{UF}\}$
Sensor 2	$S_2 \in \{\text{NF}, \text{UF}\}$

(11.3)

The system behavioral modes for the four combinations of component behavioral modes are

$$\mathbf{B} = \{\mathbf{NF}, \mathbf{S_1}, \mathbf{S_2}, \mathbf{S_1}\&\mathbf{S_2}\} \tag{11.4}$$

Let a diagnosis system Δ for this model is defined by

Δ	$H_i^0 : \Phi_i$	M_i	T_i	$\mathcal{R}_i$
δ_1	$\{\mathbf{NF}\}$	$\{e_1, e_2, e_3\}$	$z_1 - z_2^2$	$\mathbb{R}\backslash\{0\}$
δ_2	$\{\mathbf{S_2}\}$	$\{e_1, e_2\}$	z_1	$\mathbb{R}_-$

(11.5)

We will show that $\mathbf{S_1}\&\mathbf{S_2}$ is detectable with Δ, i.e.,

$$(\mathbf{S_1}\&\mathbf{S_2}, \mathbf{NF}) \in I(\Delta) \tag{11.6}$$

Let $z_0 = (z_1, z_2) = (5, 0)$ then

$$T_1(z_0) = 5 \in \mathcal{R}_1 = \mathbb{R}\backslash\{0\} \tag{11.7}$$

and

$$T_2(z_0) = 5 \notin \mathcal{R}_2 = \mathbb{R}_- \tag{11.8}$$

The candidate set is then

$$C(z_0) = \bigcap_{H_i^0 \text{ rejected}} \Phi_i^C = \Phi_1^C = \{\mathbf{S_1}, \mathbf{S_2}, \mathbf{S_1}\&\mathbf{S_2}\} \tag{11.9}$$

The fault mode $\mathbf{S_1}\&\mathbf{S_2}$ is a candidate because $\mathbf{S_1}\&\mathbf{S_2} \in C(z_0)$ and $\mathbf{NF}$ is not a candidate because $\mathbf{NF} \notin C(z_0)$. Thus, according to Definition 11.3, the double fault $\mathbf{S_1}\&\mathbf{S_2}$ is detectable with Δ.

11.2 Problem Formulation

From the discussion in the introduction of this chapter, we know that it can be especially important to isolate some specific faults from some other faults. This knowledge can be described with a binary relation I_d defined on $\mathbf{B} \times \mathbf{B}$, where

$$(b_1, b_2) \in I_d \tag{11.10}$$

if and only if it is desired that b_1 is isolable from b_2. The relation I_d will be called the *desired isolability*.

Assume that there is a set Δ_a of all possible tests, one for each rejectable model, that has been found. The test assumptions for these tests are given by $\Phi_i = \text{assump}(M_i)$. Given a desired isolability, the isolability analysis that will be presented in this chapter can be used to answer the following questions:

- Is it possible to obtain the desired isolability using all possible tests in Δ_a?
- If so, how is a subset of tests $\Delta \subseteq \Delta_a$ selected such that the desired isolability is obtained?
- If the desired isolability cannot be obtained with Δ_a, which isolability properties are missing?
- How can a small subset $\Delta \subseteq \Delta_a$ of tests be selected with the maximum isolability?

Computing all potential tests in Δ_a can be difficult or even impossible and also unnecessary as we will see later. Here we propose a method that only uses the test assumptions Φ_i to select tests. Then only for the selected tests the analytical properties, i.e., the test quantities and the rejection regions have to be derived.

11.3 Predicting the Isolability

The selection of tests is based on an isolability prediction. The isolability prediction uses, as said above, only the test assumptions, and is defined as follows.

Definition 11.4 (Structural Isolability of a Diagnosis System)**.** *Let Δ be a diagnosis system. A binary relation $I_s(\Delta)$ on $\mathbf{B} \times \mathbf{B}$ is the **structural isolability of a diagnosis system** Δ if*

$$I_s(\Delta) = \{(b_1, b_2) | \exists \delta_i \in \Delta : \left(b_1 \notin \Phi_i \wedge b_2 \in \Phi_i\right)\} \tag{11.11}$$

If $(b_1, b_2) \in I_s(\Delta)$ we say that b_1 is *structurally isolable* from b_2 with Δ. The idea of Definition (11.11) is that if a fault b_1 is isolable from another fault b_2 with a test, then b_1 must be structurally isolable from b_2 with the test. An example will show how the definition is applied to a diagnosis system.

Example 11.2

Consider the diagnosis model (11.2) and the diagnosis system (11.5). It was shown in Example 11.1 that

$$(\mathbf{S_1}\&\mathbf{S_2}, \mathbf{NF}) \in I(\Delta)$$

Now, we will determine if

$$(\mathbf{S_1}\&\mathbf{S_2}, \mathbf{NF}) \in I_s(\Delta) \tag{11.12}$$

by using expression (11.11). Since

$$\mathbf{S_1}\&\mathbf{S_2} \notin \Phi_1 \tag{11.13}$$

and

$$\mathbf{NF} \in \Phi_1 \tag{11.14}$$

it follows that δ_1 together with (11.11) imply (11.12). Hence (11.12) is true, i.e., $\mathbf{S_1}\&\mathbf{S_2}$ is structurally isolable from **NF** with Δ.

In Example 11.1 and in the previous example, it turned out that the behavioral mode $\mathbf{S_1}\&\mathbf{S_2}$ is both analytically and structurally isolable from **NF** with the diagnosis system in (11.5). Even if $(\mathbf{S_1}\&\mathbf{S_2}, \mathbf{NF})$ was contained in both isolability relations, equality between structural and analytical isolability is in general not true and this will be exemplified later. However, the following theorem shows that the structural isolability is an upper bound for the analytical isolability.

Theorem 11.1. *Given a diagnosis system Δ, it holds that*

$$I(\Delta) \subseteq I_s(\Delta) \tag{11.15}$$

Proof. Take an arbitrary $(b_1, b_2) \in I(\Delta)$. From Definition 11.3, it follows that there exists a $z = z_0$ such that b_1 is a candidate and b_2 is not a candidate, i.e.,

$$b_1 \in C(z_0) \tag{11.16}$$

and

$$b_2 \notin C(z_0) \tag{11.17}$$

From (2.20) and (11.17), it follows that

$$b_2 \notin \bigcap_{H_i^0 \text{ rejected}} \Phi_i^C \tag{11.18}$$

This means that there is a test δ_1 such that H_1^0 is rejected and

$$b_2 \notin \Phi_1^C \tag{11.19}$$

or equivalently

$$b_2 \in \Phi_1 \tag{11.20}$$

From (2.20) and (11.16), it follows that

$$b_1 \in \bigcap_{H_i^0 \text{ rejected}} \Phi_i^C \tag{11.21}$$

and it means that

$$b_1 \in \Phi_i^C \tag{11.22}$$

or equivalently

$$b_1 \notin \Phi_i \tag{11.23}$$

for all δ_i such that H_i^0 is rejected. Hence it holds also for δ_1. From (11.20), (11.23) where i is substituted for 1, and (11.11), it follows that $(b_1, b_2) \in I_s(\Delta)$. Since (b_1, b_2) was an arbitrarily chosen in $I(\Delta)$, the theorem follows. □

This theorem states that the structural isolability is necessary condition for the analytical isolability. We can also say that the structural isolability is an upper bound for the analytical isolability. Thus if a behavioral mode b_1 is not structurally isolable from another behavioral mode b_2, then b_1 is also not analytically isolable from b_2. Hence by computing the structural isolability, it is possible to conclude that behavioral modes are not analytical isolable from other behavioral modes.

11.4 Computing Structural Isolability

We have seen that information about the analytical isolability can be obtained from the structural isolability. In this section, the structural isolability is rewritten on a form suitable for computation. The structural isolability on this form is obtained by computing the structural isolability of each test in a diagnosis system Δ. The structural isolability of a single test δ_i is defined as

$$I_s(\{\delta_i\}) = \{(b_1, b_2) | b_1 \notin \Phi_i \wedge b_2 \in \Phi_i\} \tag{11.24}$$

according to (11.11). The next theorem shows how the structural isolability of a diagnosis system can then be expressed by using the structural isolabilities of its tests.

Theorem 11.2. *Given a diagnosis system Δ, it holds that*

$$I_s(\Delta) = \bigcup_{\delta_i \in \Delta} I_s(\{\delta_i\}) \tag{11.25}$$

Proof. The theorem is proved by observing that

$$\begin{aligned} I_s(\Delta) = &\ \{(b_1, b_2) | \exists \delta_i \in \Delta : \left(b_1 \notin \Phi_i \wedge b_2 \in \Phi_i\right)\} = \\ = &\ \{(b_1, b_2) | \textstyle\bigvee_{\delta_i \in \Delta} \left(b_1 \notin \Phi_i \wedge b_2 \in \Phi_i\right)\} = \\ = &\ \textstyle\bigcup_{\delta_i \in \Delta} \{(b_1, b_2) | b_1 \notin \Phi_i \wedge b_2 \in \Phi_i\} = \\ = &\ \textstyle\bigcup_{\delta_i \in \Delta} I_s(\{\delta_i\}) \end{aligned}$$

□

This expression of structural isolability clearly shows how each test contributes to the structural isolability. The structural isolability $I_s(\Delta)$ of a diagnosis system Δ can be computed by using (11.24) for each test $\delta_i \in \Delta$ and (11.25). These computations are exemplified next.

Example 11.3

Consider the diagnosis system (11.5). For this diagnosis system, it holds that $\Phi_1 = \{\mathbf{NF}\}$ and $\Phi_2 = \{\mathbf{S_2}\}$. The structural isolability of the first test is

$$\begin{aligned} I_s(\{\delta_1\}) = &\ \{(b_1, b_2) | b_1 \in \mathbf{B} \backslash \{\mathbf{NF}\} \wedge b_2 = \mathbf{NF}\} = \\ = &\ \{(\mathbf{S_1}, \mathbf{NF}), (\mathbf{S_2}, \mathbf{NF}), (\mathbf{S_1}\&\mathbf{S_2}, \mathbf{NF})\} \end{aligned} \tag{11.26}$$

The structural isolability of the test δ_2 is

$$\begin{aligned} I_s(\{\delta_2\}) = &\ \{(b_1, b_2) | b_1 \in \mathbf{B} \backslash \{\mathbf{S_2}\} \wedge b_2 = \mathbf{S_2}\} = \\ = &\ \{(\mathbf{NF}, \mathbf{S_2}), (\mathbf{S_1}, \mathbf{S_2}), (\mathbf{S_1}\&\mathbf{S_2}, \mathbf{S_2})\} \end{aligned} \tag{11.27}$$

Now, the structural isolability of the diagnosis system $\Delta = \{\delta_1, \delta_2\}$ is

$$\begin{aligned} I_s(\Delta) = \;& I_s(\{\delta_1\}) \cup I_s(\{\delta_2\}) = \\ = \;& \{(\mathbf{S_1}, \mathbf{NF}), (\mathbf{S_2}, \mathbf{NF}), (\mathbf{S_1}\&\mathbf{S_2}, \mathbf{NF}), \\ & (\mathbf{NF}, \mathbf{S_2}), (\mathbf{S_1}, \mathbf{S_2}), (\mathbf{S_1}\&\mathbf{S_2}, \mathbf{S_2})\} \end{aligned} \tag{11.28}$$

according to (11.25).

11.5 Isolability Properties and Representations

As seen in the previous example, the set representation of $I_s(\Delta)$ is difficult to interpret even for small size systems like (11.5). In this section, we will investigate different representations of the two isolability relations. We will do this by studying $I_s(\Delta)$ and the representations for the analytical correspondence $I(\Delta)$ follow analogously.

To obtain a compact representation, we will consider the relation defined as the complement set to $I_s(\Delta)$ on $\mathbf{B} \times \mathbf{B}$. This relation will be called the structural candidate implication relation and will be denoted by $\overline{I_s(\Delta)}$. If for example $(b_1, b_2) \in \overline{I_s(\Delta)}$, it will be show that b_1 is not isolable from b_2 with Δ. This means that b_2 is a candidate if b_1 is a candidate and this motivates the name of $\overline{I_s(\Delta)}$.

A common representation of a *relation* R is to use a *relation matrix* $R_m = (r_{ij})$. If R is a relation on a finite set $\mathbf{B} \times \mathbf{B}$, then

$$r_{ij} = \begin{cases} 1 \text{ if } (b_i, b_j) \in R \\ 0 \text{ if } (b_i, b_j) \notin R \end{cases} \tag{11.29}$$

The candidate implication relations can then be represented with the their relation matrix respectively. To make the candidate implication matrix even easier to interpret, the ones are replaced with "X" and the zeros are left out. The interpretation of an "X" in position (i, j) is that for all different z, b_j is a candidate if b_i is a candidate. Hence b_i is not analytically isolable from b_j with Δ. The structural candidate implication matrix of $I_s(\Delta)$ in (11.28) is

present	necessary interpreted mode			
mode	**NF**	$\mathbf{S_2}$	$\mathbf{S_1}$	$\mathbf{S_1}\&\mathbf{S_2}$
NF	X		X	X
$\mathbf{S_2}$		X	X	X
$\mathbf{S_1}$			X	X
$\mathbf{S_1}\&\mathbf{S_2}$			X	X

(11.30)

Note that the candidate implication matrix shows the complement set of the isolability relation. The isolability relation for (11.30) is the set corresponding to the blank entries in the isolability matrix.

A more compact representation of the candidate implication relation can be obtained by forming a partial order. A relation is partial order if it is reflexive, transitive, and antisymmetric. To define a partial order relation, note first that $\overline{I_s(\Delta)}$ is both *reflexive* and *transitive*. By modifying this relation according to the following definition, it is possible to obtain a relation that is also *antisymmetric*, i.e., a relation that is a partial order.

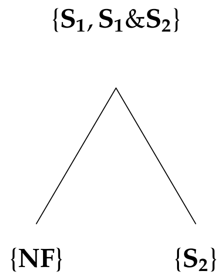

Figure 11.1: The Hasse diagram of $P_s(\Delta)$ of (11.30).

Definition 11.5 (Candidate Implication Partial Order)**.** *For a binary relation $I_s(\Delta)$ on $\mathbf{B} \times \mathbf{B}$, let $\mathbf{B}'$ be a partition of $\mathbf{B}$ defined as the set of the equivalence classes of $\overline{I_s(\Delta)}$) on $\mathbf{B}$. Further, let the equivalence class that contains b be denoted by $[b]$. Then the **structural candidate implication partial order** $P_s(\Delta)$ on $\mathbf{B}' \times \mathbf{B}'$ is defined as*

$$([b_1], [b_2]) \in P_s(\Delta) \leftrightarrow (b_1, b_2) \in \overline{I_s(\Delta)} \tag{11.31}$$

The analytical candidate implication partial order $P(\Delta)$ on $\mathbf{B}' \times \mathbf{B}'$ can be defined analogously. The relation matrix of $P_s(\Delta)$ defined by (11.30) is

present	necessary interpreted modes		
modes	$\{\mathbf{NF}\}$	$\{\mathbf{S_2}\}$	$\{\mathbf{S_1}, \mathbf{S_1}\&\mathbf{S_2}\}$
$\{\mathbf{NF}\}$	X		X
$\{\mathbf{S_2}\}$		X	X
$\{\mathbf{S_1}, \mathbf{S_1}\&\mathbf{S_2}\}$			X

(11.32)

and it can be seen that this representation is more compact compared to the one in (11.30).

A partial order P on a set $\mathbf{B} \times \mathbf{B}$ can be represented by a *Hasse diagram*, see (Skiena, 1990; Råde and Westergren, 1992). The Hasse diagram for $P_s(\Delta)$ in (11.32) is shown in Figure 11.1. In such a diagram each equivalence class is represented with a node. If $([b_1], [b_2]) \in P$ where $b_1, b_2 \in \mathbf{B}$ and $[b_1] \neq [b_2]$, then b_1 is at a lower level then b_2, and there exists a path from b_1 up-wards to b_2. Next, we will describe the interpretations of the candidate implication partial orders and then also discuss the interpretation of the Hasse diagram in Figure 11.1.

The analytical implication partial order $P(\Delta)$ has a nice interpretation that follows from Definition 11.3 and Definition 11.5, i.e., if b_1 is a candidate then all b_2 that fulfill

$$([b_1], [b_2]) \in P_s(\Delta) \tag{11.33}$$

are candidates.

The next corollary of Theorem 11.1 shows that the structural implication partial order can be interpreted in the same way. In the corollary we need to explicitly state that a relation R is used to obtain the equivalence class $[b]$ and then the notation $[b]_R$ will be used.

Corollary 11.1. *Given a diagnosis system* Δ, *it holds that*

$$([b_1]_{\overline{I_s(\Delta)}}, [b_2]_{\overline{I_s(\Delta)}}) \in P_s(\Delta) \tag{11.34}$$

implies

$$([b_1]_{\overline{I(\Delta)}}, [b_2]_{\overline{I(\Delta)}}) \in P(\Delta) \tag{11.35}$$

Proof. Take an arbitrary

$$([b_1]_{\overline{I_s(\Delta)}}, [b_2]_{\overline{I_s(\Delta)}}) \in P_s(\Delta) \tag{11.36}$$

From (11.31) and (11.36), it follows that

$$(b_1, b_2) \in \overline{I_s(\Delta)} \tag{11.37}$$

By using Theorem 11.1 and (11.37), we get that

$$(b_1, b_2) \in \overline{I(\Delta)} \tag{11.38}$$

Finally (11.38) and (11.31) imply (11.35). □

Corollary 11.1 implies that if b_1 is a candidate then all b_2 that fulfill

$$([b_1], [b_2]) \in P_s(\Delta) \tag{11.39}$$

are candidates. In (11.33) we used the analytical properties of Δ to draw conclusions about the isolation capability but in (11.39) only the test assumptions Φ_i are used. In Figure 11.1 it can be seen for example that, if $\mathbf{S_2}$ is a candidate, then $\mathbf{S_1}$ and $\mathbf{S_1}\&\mathbf{S_2}$ are candidates too according to Corollary 11.1.

11.6 Analytical Isolability Properties

In Section 11.4 we showed that the structural isolability of a diagnosis system is equal to the union of structural isolability of each test. In this section we investigate the relationship between the analytical isolability of a diagnosis system $I(\Delta)$ and the union of the analytical isolability of each test in Δ. The analytical isolability of a test δ_i is

$$I(\{\delta_i\}) = \{(b_1, b_2) | (\exists z \in \mathcal{Z} : H_i^0 \text{ rejected}) \wedge b_1 \notin \Phi_i \wedge b_2 \in \Phi_i\} \tag{11.40}$$

according to Definition 11.3. By comparing this expression with (11.24), it follows that if there is a z such that H_i^0 is rejected, then

$$I(\{\delta_i\}) = I_s(\{\delta_i\}) \tag{11.41}$$

Later we will show that the analytical isolability can, contrary to the structural isolability, in general not be computed as the union of the isolability properties obtained from each test. However the next theorem shows that the union of isolability properties obtained from each test is a superset to the analytical isolability. Note that we do not need to assume that (11.41) holds, to obtain the following result.

Lemma 11.1. *Given a diagnosis system* Δ*, it holds that*

$$I(\Delta) \subseteq \bigcup_{\delta_i \in \Delta} I(\{\delta_i\}) \tag{11.42}$$

Proof. Let $\Delta = \Delta_r \cup \Delta_n$ be a partition. A test $\delta_i \in \Delta$ is contained in Δ_r if and only if there exists an observation such that its null hypothesis can be rejected. This means that the null hypothesis for the tests in Δ_n cannot be rejected for any observation and therefore does not influence the candidate set. From this and (11.1), it follows that

$$I(\Delta) = I(\Delta_r \cup \Delta_n) = I(\Delta_r) \tag{11.43}$$

From Theorem 11.1, we get that

$$I(\Delta_r) \subseteq I_s(\Delta_r) \tag{11.44}$$

The structural isolability can be expressed as

$$I_s(\Delta_r) = \cup_{\delta_i \in \Delta_r} I_s(\{\delta_i\}) \tag{11.45}$$

according to (11.25). The structural and the analytical isolabilities are equal for all tests in Δ_r according to (11.41). The right-hand side of (11.45) can then be written as

$$\cup_{\delta_i \in \Delta_r} I_s(\{\delta_i\}) = \cup_{\delta_i \in \Delta_r} I(\{\delta_i\}) \tag{11.46}$$

Finally we have that

$$\cup_{\delta_i \in \Delta_r} I(\{\delta_i\}) \subseteq \cup_{\delta_i \in \Delta} I(\{\delta_i\}) \tag{11.47}$$

and the theorem follows from (11.43)-(11.47). □

The right-hand side of (11.42) is in the normal case equal to the structural isolability $I_s(\Delta)$ and this can be realized as follows. As argued Section 2.3.3, almost always there exists an observation z_0 such that $T_i(z_0) \in \mathcal{R}_i$, i.e., (11.41) is satisfied. If all tests in a diagnosis system has this property, then it follows directly from (11.25) and (11.41) that the structural isolability of the diagnosis system is equal to the union of the analytical isolability of each test, i.e.,

$$I_s(\Delta) = \bigcup_{\delta_i \in \Delta} I_s(\{\delta_i\}) = \bigcup_{\delta_i \in \Delta} I(\{\delta_i\}) \tag{11.48}$$

Hence in this case, it follows that $I(\Delta) \neq \bigcup_{\delta_i \in \Delta} I(\{\delta_i\})$ if and only if $I(\Delta) \neq I_s(\Delta)$ and inequality in the latter expression will be investigated in the next section.

11.7 Comparison between Structural and Analytical Isolability

From Theorem 11.1, we know that structural isolability is a necessary condition for analytical isolability. In this section we use Example 11.3 to show that inequality between the structural isolability and the analytical isolability can hold. For the diagnosis system in Example 11.3, it is possible to calculate $I(\Delta)$. Before the calculations are carried out, a useful definition is presented.

Definition 11.6 (Acceptance Set)**.** *Given a diagnosis test δ, the **acceptance set** for δ is*

$$\mathcal{O}(\delta) := \{z \in \mathcal{Z} | T(z) \in \mathcal{R}^C\} \tag{11.49}$$

Theoretically, $\mathcal{O}(\delta)$ contains equivalent information as T and $\mathcal{R}$. However in practice, T and $\mathcal{R}$ also express an efficient way to evaluate δ. In the next example $I(\Delta)$ is computed for (11.5) and compared with $I_s(\Delta)$ which has already been computed in Example 11.3.

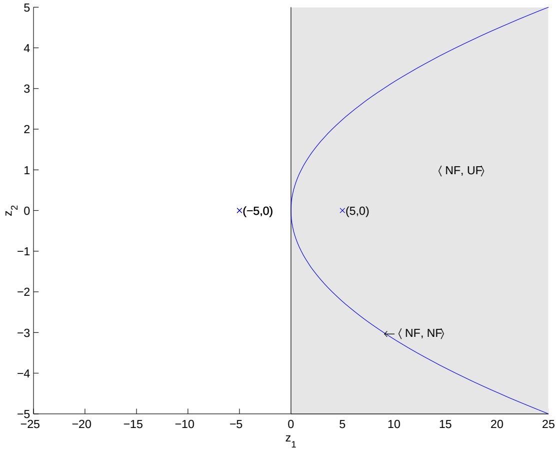

Figure 11.2: The parabola $\mathcal{O}(\delta_1)$ and the closed right half plane $\mathcal{O}(\delta_2)$ shown in the z-plane.

Example 11.4

Consider again the model introduced in Example 11.1. One way to illustrate a static diagnosis system with two known variables is to plot the acceptance set of each test. For the diagnosis system (11.5) this plot is shown in Figure 11.2. The set $\mathcal{O}(\delta_1)$ is the parabola and $\mathcal{O}(\delta_2)$ is right half-plane. The set $\mathcal{O}(\delta_1)$ and $\mathcal{O}(\delta_2)$ defines a partition of the space of observations into 3 parts, i.e.,

$$\mathcal{O}(\delta_1) \cap \mathcal{O}(\delta_2) \tag{11.50}$$

and

$$\mathcal{O}(\delta_2) \setminus \mathcal{O}(\delta_1) \tag{11.51}$$

and

$$\mathcal{O}(\delta_1)^C \cap \mathcal{O}(\delta_2)^C \tag{11.52}$$

Observations in each of these three sets imply a different set of candidates. From the definition of structural isolability, it follows that for a given z, we have that

$$\{(b_1, b_2) | b_1 \in \mathcal{C}(z) \wedge b_2 \notin \mathcal{C}(z)\} \subseteq I(\Delta) \tag{11.53}$$

If an observation belongs to the set (11.50) then no null hypothesis is rejected and hence all behavioral modes are candidates. Since all behavioral modes are candidates, no isolability property in $I(\Delta)$ is implied. If an observation belongs

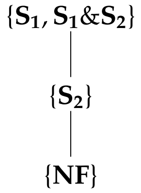

Figure 11.3: Hasse diagram of (11.57).

to the set (11.51) then H_1^0 is rejected and the result is that all behavioral modes except for **NF** are candidates. According to the definition of $I(\Delta)$, it follows that

$$\begin{aligned}
&\{(b_1, b_2)|b_1 \in C(z) \wedge b_2 \notin C(z)\} = \\
&\{(b_1, b_2)|b_1 \in \mathbf{B}\backslash\{\mathbf{NF}\} \wedge b_2 = \mathbf{NF}\} = \\
&\{(\mathbf{S_1}, \mathbf{NF}), (\mathbf{S_2}, \mathbf{NF}), (\mathbf{S_1}\&\mathbf{S_2}, \mathbf{NF})\} \subseteq I(\Delta)
\end{aligned} \tag{11.54}$$

Finally if an observation belongs to the set (11.52) holds then both null hypothesis are rejected and the candidates are $\mathbf{S_1}$ and $\mathbf{S_1}$&$\mathbf{S_2}$. This case implies that

$$\begin{aligned}
&\{(b_1, b_2)|b_1 \in \{\mathbf{S_1}, \mathbf{S_1}\&\mathbf{S_2}\} \wedge b_2 \in \{\mathbf{NF}, \mathbf{S_2}\}\} = \\
&\{(\mathbf{S_1}, \mathbf{NF}), (\mathbf{S_1}\&\mathbf{S_2}, \mathbf{NF}), (\mathbf{S_1}, \mathbf{S_2}), (\mathbf{S_1}\&\mathbf{S_2}, \mathbf{S_2})\} \subseteq I(\Delta)
\end{aligned} \tag{11.55}$$

Since only the three discussed candidates can be obtained for any observation, it follows that $I(\Delta)$ is the union of the set in (11.54) and set in (11.55), i.e.,

$$I(\Delta) = \{(\mathbf{S_1}, \mathbf{NF}), (\mathbf{S_2}, \mathbf{NF}), (\mathbf{S_1}\&\mathbf{S_2}, \mathbf{NF}), (\mathbf{S_1}, \mathbf{S_2}), (\mathbf{S_1}\&\mathbf{S_2}, \mathbf{S_2})\} \tag{11.56}$$

The analytical candidate implication matrix of $I(\Delta)$ is

$$\begin{array}{l|cccc}
\text{present} & \multicolumn{4}{l}{\text{necessary interpreted mode}} \\
\text{mode} & \mathbf{NF} & \mathbf{S_2} & \mathbf{S_1} & \mathbf{S_1}\&\mathbf{S_2} \\
\hline
\mathbf{NF} & X & \mathbf{X} & X & X \\
\mathbf{S_2} & & X & X & X \\
\mathbf{S_1} & & & X & X \\
\mathbf{S_1}\&\mathbf{S_2} & & & X & X
\end{array} \tag{11.57}$$

and the corresponding Hasse diagram is shown in Figure 11.3. A comparison of $I_s(\Delta)$ and $I(\Delta)$ reveals that

$$I_s(\Delta) = I(\Delta) \cup \{(\mathbf{NF}, \mathbf{S_2})\} \tag{11.58}$$

This difference is marked with a bold "**X**" in (11.57).

Note that in the previous example the analytical isolability is a proper subset of the structural isolability, i.e., $I(\Delta) \subset I_s(\Delta)$. From (11.26) and (11.27) we see that if there is an observation such that only the null hypothesis of δ_2 is rejected, then $(\mathbf{NF}, \mathbf{S_2})$ would be included in $I(\Delta)$. However,

$$O(\delta_1) \subset O(\delta_2) \tag{11.59}$$

implies that the null hypotheses of test δ_1 is rejected when the null hypothesis of test δ_2 is rejected. Hence the tests have analytical constraints of which sets of tests that can be invalidated. In the structural isolability these constraints are not considered and this implies that the structural isolability can be more optimistic than the analytical isolability.

11.8 Sufficient Condition for Equality

In the previous example, it was shown that there exist a diagnosis systems Δ such that $I_s(\Delta) \neq I(\Delta)$. In this section we present a sufficient condition for equality between the structural and the analytical isolability. This condition is based on the influence of faults. A behavioral mode influence a test δ_i if there exists an observation $z \in \mathcal{O}(M_b)$ such that

$$T_i(z) \in \mathcal{R}_i \tag{11.60}$$

It has been shown in Section 2.2.1 that it is desirable to chose Φ_i^C equal to the fault influence of the test to obtain the strongest conclusions when H_i^0 is rejected. Therefore it is reasonable to compute exactly which faults that influence each test. This can be done for example by analytical computations, by Monte Carlo-simulations, or by measurement data from a faulty process. However, this will not be the topic here. If Φ_i^C is equal to the fault influence then it will be shown that it is straightforward to compute the analytical isolability of the diagnosis system. First, we show that the analytical isolability is equal to the union of the analytical isolabilities of each test under this condition.

Lemma 11.2. *Given a diagnosis system Δ where for each $\delta_i \in \Delta$, the set Φ_i^C is equal to the set of behavioral modes that influences δ_i, it holds that*

$$I(\Delta) = \bigcup_{\delta_i \in \Delta} I(\{\delta_i\}) \tag{11.61}$$

Proof. Lemma 11.1 implies that it remains to prove that

$$\bigcup_{\delta_i \in \Delta} I(\{\delta_i\}) \subseteq I(\Delta) \tag{11.62}$$

This is shown by proving that for an arbitrary test $\delta_i \in \Delta$, it follows that

$$I(\{\delta_i\}) \subseteq I(\Delta) \tag{11.63}$$

Let $\delta_0 \in \Delta$ be an arbitrarily chosen test. If $I(\{\delta_0\}) = \emptyset$, then (11.63) follows trivially. If $I(\{\delta_0\}) \neq \emptyset$, then let (b_1, b_2) be an arbitrary element in $I(\{\delta_0\})$.

Since $(b_1, b_2) \in I(\{\delta_0\})$, it follows from (11.40) that $b_1 \in \Phi_0^C$. From the definition of fault influence and the fact that $b_1 \in \Phi_0^C$ it follows that there exists a $z = z_0 \in \mathcal{O}(M_{b_1})$ such that

$$z_0 \notin \mathcal{O}(\delta_0) \tag{11.64}$$

From $z_0 \in \mathcal{O}(M_{b_1})$ and the definition of diagnosis, it follows that $b_1 \in \mathcal{D}(z_0)$. Only tests influenced by b_1 can be rejected at z_0, and for all such tests δ_i it holds that $b_1 \in \Phi_i^C$. This and (2.20) imply that

$$b_1 \in C(z_0) \tag{11.65}$$

Since $(b_1, b_2) \in I(\{\delta_0\})$, it follows from (11.40) that $b_2 \in \Phi_0$. For z_0, it follows that $\Phi_0^C \supseteq C(z_0)$ according to (2.20). This and the fact that $b_2 \in \Phi_0$ imply that

$$b_2 \notin C(z_0) \tag{11.66}$$

Now, by combining (11.65) and (11.66), we get that $(b_1, b_2) \in I(\{\Delta\})$. Since (b_1, b_2) was arbitrarily chosen in $I(\{\delta_0\})$, it follows that (11.63) holds. Since $\delta_0 \in \Delta$ was arbitrarily chosen, (11.62) follows and then also the theorem. □

Next, we show that the structural and analytical isolability is equal under this condition. Note that we do not need to assume that (11.48) holds, to obtain the following result.

Theorem 11.3. *Given a diagnosis system Δ where for each $\delta_i \in \Delta$ the set Φ_i^C is equal to the set of behavioral modes that influences δ_i, it follows that*

$$I(\Delta) = I_s(\Delta) \tag{11.67}$$

Proof. Let δ_i be an arbitrarily chosen test in Δ. If there exist no observation such that H_i^0 is rejected, it follows that δ_i is influenced by no behavioral mode, i.e., $\Phi_i^C = \emptyset$ and $\Phi_i = \mathbf{B}$. This gives according to (11.40) and (11.24) that

$$I(\{\delta_i\}) = I_s(\{\delta_i\}) = \emptyset \tag{11.68}$$

If there exists an observation such that H_i^0 is rejected, then (11.41) holds. Hence, it follows from (11.68) and (11.41) that

$$I(\{\delta_i\}) = I_s(\{\delta_i\}) \tag{11.69}$$

for all $\delta_i \in \Delta$.

Now, Theorem 11.2 gives that

$$I_s(\Delta) = \bigcup_{\delta_i \in \Delta} I_s(\{\delta_i\}) \tag{11.70}$$

Since (11.69) holds for all $\delta_i \in \Delta$, it follows that

$$\bigcup_{\delta_i \in \Delta} I_s(\{\delta_i\}) = \bigcup_{\delta_i \in \Delta} I(\{\delta_i\}) \tag{11.71}$$

The condition in Lemma 11.2 is fulfilled and it follows that

$$\bigcup_{\delta_i \in \Delta} I(\{\delta_i\}) = I(\Delta) \tag{11.72}$$

The equalities (11.70)-(11.72) imply (11.67) and this completes the proof. □

Next this theorem will be illustrated.

Example 11.5

Consider Example 11.4. In (11.59), it was concluded that

$$\mathcal{O}(M_{\mathbf{NF}}) \subset \mathcal{O}(\delta_2) \tag{11.73}$$

This means that δ_2 is not influenced by **NF** and it should therefore be included in Φ_2, i.e.,

$$\Phi_2 = \{\mathbf{NF}, \mathbf{S_2}\} \tag{11.74}$$

to fulfill the condition in Theorem 11.3. To distinguish the modified test and the resulting diagnosis system from the original test δ_2 and the original diagnosis system Δ, let the new test be denoted by δ_2' and let the new diagnosis system be denoted by $\Delta' = \{\delta_1, \delta_2'\}$. Next the structural isolability of the new diagnosis system Δ' will be computed by using (11.25). The structural isolability $I_s(\delta_1)$ is (11.26). The structural isolability of $I_s(\delta_2')$ is

$$\begin{aligned} I_s(\delta_2') = & \{(b_1, b_2)|b_1 \in \{\mathbf{S_1}, \mathbf{S_1\&S_2}\} \wedge b_2 \in \{\mathbf{NF}, \mathbf{S_2}\}\} = \\ = & \{(\mathbf{S_1}, \mathbf{NF}), (\mathbf{S_1\&S_2}, \mathbf{NF}), (\mathbf{S_1}, \mathbf{S_2}), (\mathbf{S_1\&S_2}, \mathbf{S_2})\} \end{aligned} \tag{11.75}$$

The corresponding expression to (11.28) is

$$\begin{aligned} I_s(\Delta') = & I_s(\delta_1) \cup I_s(\delta_2') = \\ = & \{(\mathbf{S_1}, \mathbf{NF}), (\mathbf{S_2}, \mathbf{NF}), (\mathbf{S_1\&S_2}, \mathbf{NF}), (\mathbf{S_1}, \mathbf{S_2}), (\mathbf{S_1\&S_2}, \mathbf{S_2})\} \end{aligned} \tag{11.76}$$

According to Theorem 11.3 we get that $I(\Delta') = I_s(\Delta')$. A comparison between the two different structural isolability relations gives that

$$I(\Delta') = I_s(\Delta') \subset I_s(\Delta) \tag{11.77}$$

In conclusion, the structural isolability is an upper bound for analytical isolability. Equality is obtained when Φ_i is chosen as the largest possible set, i.e., when all faults in Φ_i^C influence the corresponding test. If the exact set of behavioral modes that influence each test is unknown then the following principle can be used to get the best prediction of the analytical isolability. For best prediction, the sets Φ_i should be chosen as the largest set of behavioral modes that do not influence the corresponding tests.

11.9 Test Selection Based on Isolability Analysis

In this section, we will answer the four questions posed in Section 11.2 by using the theoretical tools developed in this chapter. All these questions concern the set of potential tests Δ_a and this set can be obtained as follows.

Rejectable models can be found using structural analysis and tests can be based on rejectable models. Given a rejectable model M_i, the test assumption $\Phi_i = \text{assump}(M_i)$ is easily computed, but a test quantity T_i and a rejection region $\mathcal{R}_i$ might be difficult and time-consuming to derive. Therefore we will only use the test assumptions for selecting some of the potential tests. In this way, we only need to derive test quantities and rejection regions for the selected tests. The set of all possible tests δ_i is Δ_a. Note once again that only the sets Φ_i are explicitly given for each possible test in Δ_a.

11.9.1 Necessary Condition for Desired Isolability

The first of the four questions is if it possible to obtain the desired isolability I_d using all possible tests in Δ_a? By using only the sets Φ_i of the tests in Δ_a, the structural isolability $I_s(\Delta_a)$ can be computed by using (11.25). It holds that $I(\Delta_a) \subseteq I_s(\Delta_a)$, according to Theorem 11.1. If it holds that

$$I_d \subseteq I_s(\Delta_a) \tag{11.78}$$

then it is possible, but not sure, that the desired isolability can be obtained, i.e.,

$$I_d \subseteq I(\Delta_a) \tag{11.79}$$

11.9.2 Test Selection

The second question is if (11.78) is true, how is a minimum cardinality subset of tests $\Delta \subseteq \Delta_a$ selected such that the desired isolability is obtained? In this case, a minimum cardinality subset Δ of the potential tests in Δ_a can be selected such that

$$I_d \subseteq I_s(\Delta) \tag{11.80}$$

This problem is the classical set covering optimization problem that is shown to be NP-hard (Karp, 1972). Test quantities and rejection regions are derived only for the selected tests in Δ. The set of faults that influence each test in Δ is evaluated. If the set of faults that influence the tests are Φ_i^C, then it follows that

$$I_d \subseteq I_s(\Delta) = I(\Delta) \tag{11.81}$$

according to Theorem 11.3. This means that a diagnosis system Δ with the desired isolability has been derived.

If there exist some tests where the set of faults that influence the tests are a proper subset of Φ_i^C respectively, then the sets Φ_i^C can be changed to the set of faults that influence the corresponding test. This will be discussed more in Section 12.5. Theorem 11.3 can then be used to compute $I(\Delta)$ by using the new sets Φ_i. If it turns out that

$$I_d \not\subseteq I(\Delta) \tag{11.82}$$

then the new sets Φ_i can be used for the tests in $\Delta \subseteq \Delta_a$ and the selection of tests in Δ_a can be reconsidered. This can be summarized in the following design procedure:

1. Select a minimum cardinality set $\Delta \subseteq \Delta_a$ such that $I_d \subseteq I_s(\Delta)$.
2. Compute test quantities and rejection region for the tests in Δ.
3. Evaluate the set of faults that influence each test in Δ.
4. If the fault influence is not equal to Φ_i for some test in Δ, then modify the sets Φ_i for the tests in Δ_a. If $I_d \not\subseteq I_s(\Delta_a)$, then the desired isolability cannot be obtained. Else go back to 1.
5. If the fault influence is equal to Φ_i for all test in Δ, then Δ is a diagnosis system with the desired isolability.

The steps that might be difficult to perform is the second and third step. However, by minimizing the number of selected tests, we also minimize the test construction and evaluation work in these steps. By following the design procedure, a diagnosis system with the desired isolability is obtained if and only if it is possible to obtain the desired isolability with all potential tests in Δ_a.

11.9.3 Computing Missing Isolability Properties

The third question is if the desired isolability cannot be obtained with Δ_a, which isolability properties are missing? If (11.78) does not hold, i.e.,

$$I_d \not\subseteq I_s(\Delta_a) \tag{11.83}$$

then the possible tests Δ_a have *not* the desired isolability. Missing isolability properties are given by

$$I_d \backslash I(\Delta_a) \tag{11.84}$$

Since it holds that

$$I_d \backslash I_s(\Delta_a) \subseteq I_d \backslash I(\Delta_a) \tag{11.85}$$

the set $I_d \backslash I_s(\Delta)$ gives some of the missing isolability properties.

11.9.4 Obtaining Maximum Isolability

The fourth and final question was how a small subset $\Delta \subseteq \Delta_a$ of tests can be selected with the maximum isolability? One solution to this problem is the trivial solution $\Delta = \Delta_a$. There might exist solutions with less number of tests and a design procedure to find one such solution is proposed next.

The design procedure will be based on the procedure presented in Section 11.9.2. By using that procedure, a diagnosis system with the maximum isolability is obtained if the following modifications are done. The desired isolability is chosen as the structural isolability of Δ_a. Furthermore, if the fault influence of some test is modified in step (4) such that the structural isolability $I_s(\Delta_a)$ is changed, than instead of terminating we go back to step (1) and change the desired isolability to the new structural isolability. These modifications leads to the following procedure.

In the first step, the set Δ is selected such that the maximum structural isolability is obtained, i.e.,

$$I_s(\Delta) = I_s(\Delta_a) \tag{11.86}$$

The second and third steps are done as before. The fourth step is reformulated as follows. If Φ_i is not equal to the fault influence for some test in Δ, then modify Φ_i in accordance with the computed fault influence and go back to step 1. The result of the fifth step is in this case a diagnosis system with the maximum isolability and this can be realized as described next. Since the set of faults that influence each test is equal to Φ_i^C it follows that

$$I(\Delta) = I_s(\Delta) \tag{11.87}$$

This, Theorem 11.1 and (11.86) imply

$$I(\Delta_a) \subseteq I_s(\Delta_a) = I_s(\Delta) = I(\Delta) \tag{11.88}$$

i.e., no additional test in Δ_a can improve the isolability of Δ.

A difference between the procedure in Section 11.9.2 and this procedure is Step 4, where the condition $I_d \not\subseteq I_s(\Delta_a)$ has no correspondence when computing a diagnosis system with maximum isolability. This means that if all steps can be performed, the proposed design procedure will always produce a diagnosis system with the maximum isolability.

The design procedure presented here can also be used if all of the desired isolability properties can not be obtained. Then it is reasonable to find a set $\Delta \subseteq \Delta_a$ that maximizes the set $I_d \cap I(\Delta)$. Such diagnosis system is obtained if $I_d \cap I_s(\Delta)$ is used instead of $I_s(\Delta)$.

11.9.5 Example

Finally an example will show how desired isolability is defined and how conclusions are drawn using structural analysis applied to a diagnosis system.

Example 11.6
Consider the electrical circuit in Example 6.9. There are 12 MSO sets in this model and these are equal to the 12 MO sets given in (6.66). Assume that we consider one possible test for each MSO set. From Theorem 6.8 we get that the fault influence of each test is $\Phi_i = \text{assump}(M_i)$. This and Theorem 11.3 implies that the analytical isolability is equal to the structural isolability. The assumptions for the MSO sets M_i are

i	M_i	$\Phi_i = \text{assump}(M_i)$
1	$\{1,2,3,4,6\}$	$\phi(R1 = NF \wedge L2 = NF \wedge P = NF \wedge S2 = NF)$
2	$\{1,2,3,5,6\}$	$\phi(R1 = NF \wedge L2 = NF \wedge S1 = NF \wedge S2 = NF)$
3	$\{1,2,3,6,7\}$	$\phi(R1 = NF \wedge L2 = NF \wedge S2 = NF \wedge S3 = NF)$
4	$\{1,2,4,6,7\}$	$\phi(R1 = NF \wedge P = NF \wedge S2 = NF \wedge S3 = NF)$
5	$\{1,2,5,6,7\}$	$\phi(R1 = NF \wedge S1 = NF \wedge S2 = NF \wedge S3 = NF)$
6	$\{3,4,7\}$	$\phi(L2 = NF \wedge P = NF \wedge S3 = NF)$
7	$\{3,5,7\}$	$\phi(L2 = NF \wedge S1 = NF \wedge S3 = NF)$
8	$\{4,5\}$	$\phi(P = NF \wedge S1 = NF)$
9	$\{1,2,3,6,8\}$	$\phi(R1 = NF \wedge L2 = NF \wedge S2 = NF \wedge P = F)$
10	$\{1,2,6,7,8\}$	$\phi(R1 = NF \wedge S2 = NF \wedge S3 = NF \wedge P = F)$
11	$\{3,7,8\}$	$\phi(L2 = NF \wedge S3 = NF \wedge P = F)$
12	$\{5,8\}$	$\phi(S1 = NF \wedge P = F)$

(11.89)

Let the desired isolability I_d be that each single fault should be detectable and isolable from any other single fault. If the set of single faults is denoted by $\mathbf{B}_s = \{\mathbf{R1}, \mathbf{L2}, \mathbf{P}, \mathbf{S1}, \mathbf{S2}, \mathbf{S3}\}$, then the desired isolability can be written as

$$I_d = \{(b, \mathbf{NF}) | b \in \mathbf{B}_s\} \cup \{(b_1, b_2) \in \mathbf{B}_s \times \mathbf{B}_s | b_1 \neq b_2\} \tag{11.90}$$

Since (11.67) in Theorem 11.3 holds, the analytical isolability can be computed by using (11.25). The result of these computations is showed in Figure 11.4 as the part of $P(\Delta_a)$ restricted to single faults and the no fault mode. Note that there is no edge between $\{\mathbf{P}\}$ and $\mathbf{NF}$, i.e., $\mathbf{NF}$ is isolable from $\mathbf{P}$. In the figure, it can be seen that the tests Δ_a have not the capability of distinguishing the faults $\mathbf{R1}$ and $\mathbf{S2}$ or more formally that

$$I_d \setminus I(\Delta_a) = \{(\mathbf{R1}, \mathbf{S2}), (\mathbf{S2}, \mathbf{R1})\} \tag{11.91}$$

Since the desired isolability cannot be obtained, we will find a diagnosis system with as many desired isolability properties as possible, i.e., a $\Delta \subseteq \Delta_a$ such that

$$I_d \cap I(\Delta) = I_d \cap I(\Delta_a) \tag{11.92}$$

One example of a minimum cardinality set of tests with this property is $\Delta = \{\delta_1, \delta_5, \delta_6, \delta_7, \delta_8\}$. Note that this diagnosis system is not sound with respect to the behavioral mode $\mathbf{P}$. This can be seen for example by noting that no test in Δ uses the fault model in equation 8. Hence there are observations such that $\mathbf{P}$ is a candidate but not a diagnosis.

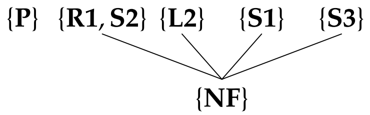

Figure 11.4: Hasse diagram of part of $P(\Delta_a)$.

11.10 Conclusions

A design procedure for test selection based on isolability properties has been proposed. Given a large set of potential tests, a small set of tests is selected with a desired isolability or the best possible isolability. Only for the selected tests, test quantities and rejection regions have to be derived. This is a big advantage since the number of potential tests can be large and to derive a test quantities for each test might be difficult. Since the test quantities of the tests are not known, the selection has been based on the structural isolability. It has been shown in Theorem 11.1 that the structural isolability is a necessary condition for the analytical isolability. This means that analytical isolability of the selected tests might be worse than predicted with the structural isolability. It has been shown in Theorem 11.3 that equality between structural and analytical isolability holds if for each selected test the set Φ_i^C is equal to the set of faults that influence the test. Hence by computing which faults that influence each selected test, it can be determined if the desired or the best possible isolability has been obtained. Finally, a design procedure that uses these ideas to find a test selection with for example the desired isolability has been given in Section 11.9.

12

I P D M A

In the previous chapter we analyzed the fault isolability of a set of tests and a procedure for test selection was described. In this chapter, the fault isolability of a diagnosis model is investigated. Isolability analysis of diagnosis models can be used for test selection and also for other purposes. In the development of processes, different design decisions are taken, e.g. how different parts are connected, which actuators to use, and which sensors to use. All these design decisions may influence the isolability possibilities. In addition, when designing the diagnosis system, there is a choice of different fault modeling strategies and which diagnosis tests to include. As a guidance when taking these design decisions, it is desirable to know how different design choices affect the isolability possibilities and this can be evaluated by studying the isolability of diagnosis models.

To find the isolability of a given model of a process is a difficult problem in general since it is related to the problem of solving large systems of non-linear differential equations. Here we will present two approaches to attack the problem. Both approaches use a structural model of a process as input and compute an isolability prediction.

In the first approach the prediction of the isolability is exact under some conditions but in general only an approximation. In the second approach we compute faults that are not isolable from other faults, i.e., an upper bound of the isolability. Since only a structural model is used, no precise analytical equations are needed. This implies that the algorithms can be used early in the design phase and thus serve as a guidance when taking different design decisions. Hence by using efficient graph theoretical methods it is possible to exclude design alternatives with insufficient isolability. However, if we need to know exactly which faults that are isolable from others, the isolability prediction also helps braking down the large problem into smaller and easier problems to analyze. We will also show that the isolability prediction in combination with the theory for test selection presented in the previous chapter can be used in the construction of a diagnosis system with the best possible isolability.

Isolability analysis has previously been studied in (Dressler and Struss, 2003), but only for qualitative models. Furthermore, a structural method for computing the isolability of different sensor configurations was presented in (Travé-Massuyès et al., 2003). This and other earlier works using structural models for diagnosis, e.g. (Pulido and Alonso, 2002), (Dustegör et al., 2006; Frisk et al., 2003), (Cassar and Staroswiecki, 1997), and (Blanke et al., 2003), have imposed analytical assumptions on the systems, e.g. that only subsystem with more equations than unknowns, i.e., only over-constrained subsystems, can be invalidated and therefore contribute to detection and isolation. This assumption is also made in the first approach that will be presented here. However these assumptions are difficult to verify in most larger models. If these assumptions are not satisfied, faults that are predicted to be isolable from other faults may be not isolable and faults that are predicted not to be isolable from other faults might be isolable. In contrast, the second approach that will be presented in this chapter does not require any analytical assumptions.

In Section 12.1 the central concepts detectability and isolability are recalled. Then we relate these concepts to structural properties of the model through the new concept of *checking model*. In the sections 12.2 and 12.3, we describe two algorithms to compute checking models by using a structural model. By combining the algorithms for finding checking models with the results relating checking models and isolability, algorithms for isolability prediction are developed in Section 12.4. An example shows how the obtained isolability prediction can be interpreted. It will be shown that for different equivalent diagnosis models, different isolability predictions can be obtained with the methods proposed here. To put the diagnosis model on a form suitable for these isolability predictions, modeling guidelines are presented in Section 12.5. Furthermore, in Section 12.6 illustrative examples show how isolability prediction can be used to identify additional fault modeling, support sensor selection to meet given isolability requirements. In Section 12.7 a design procedure for constructing a diagnosis system with the maximum possible isolability is proposed. Finally, the conclusions are drawn in Section 12.8.

12.1 Isolability of a Diagnosis Model

In this section, we will formally define and illustrate definitions concerning the fault isolation capability of a diagnosis model. We start to define two such properties of a diagnosis model.

Definition 12.1 (Isolable in a Diagnosis Model)**.** *A system behavioral-mode b_i is* ***isolable*** *from another system behavioral-mode b_j in a diagnosis model, if there exists some observation $z \in \mathbb{Z}$ such that b_i is a diagnosis but b_j is not.*

Definition 12.2 (Detectable in a Diagnosis Model)**.** *A fault mode b_i is* ***detectable*** *in a diagnosis model if b_i is isolable from the no-fault system behavioral mode in the diagnosis model.*

It could be argued that the proposed definitions are relatively weak in the following sense. For example a fault mode is detectable if there exists only one single observation that distinguish the fault from the no-fault mode. However, by using this relatively weak definition, a non-detectable fault would also be non-detectable with any stronger definition. Furthermore, for linear systems the detectability definitions in (Nyberg, 2002b) and (Nyberg and Frisk, 2006) are equivalent to the detectability definition proposed here.

To summarize which behavioral modes that are isolable from others and which faults that are detectable in a diagnosis model, the following binary relation will be used.

Definition 12.3 (Analytical Isolability of a Diagnosis Model). *Given a diagnosis model $\mathbb{M}$, a binary relation $I(\mathbb{M})$ on $\mathbf{B} \times \mathbf{B}$ is the* ***analytical isolability of the diagnosis model*** $\mathbb{M}$ *if*

$$I(\mathbb{M}) = \{(b_1, b_2) | \exists z \in \mathbb{Z} : \left(b_1 \in \mathcal{D}(z) \land b_2 \notin \mathcal{D}(z)\right)\} \tag{12.1}$$

The interpretation of $(b_1, b_2) \in I(\mathbb{M})$ is that b_1 is isolable from b_2 with the diagnosis model $\mathbb{M}$. If $(b, \mathbf{NF}) \in I(\Delta)$ for a fault mode b, then it means that b is detectable with the diagnosis model $\mathbb{M}$. For pedagogical reasons, we will first investigate how detectability can be predicted, and then generalize the results to the more general problem of how to do isolability predictions.

12.1.1 Predicting Detectability

In this section we will describe how detectability information can be derived without knowing the exact analytical equations of a model like the one in Table 12.2. It can be realized that b is not detectable if $M_{\mathbf{NF}} \subseteq M_b$. However detectability analysis by this naive idea is not particularly powerful. Here a refinement of this idea will be presented.

Consider first the no-fault system behavioral-model. As in (Blanke et al., 2003), we say that a fault can *violate* some equations in the no-fault system-behavioral model if some equations in no-fault system-behavioral model can be false for variable values consistent with the behavioral model of the fault. This will be generalized to models containing fault models and for generality we will talk about system behavioral modes instead of faults in the next definition.

Definition 12.4 (Violate). *A system behavioral mode b can* ***violate*** *equation e if*

$$\exists (x, z) \in \mathbb{X} \times \mathbb{Z} : (M_b(x, z) \land \neg e(x, z)) \tag{12.2}$$

Note that no equation in M_b can be violated by b.

Example 12.1

Throughout the chapter, we will exemplify concepts and techniques on the water-tank system presented in Example 2.1 and depicted in Figure 2.1. A model of the process in shown in Table 2.2. This model is rewritten, according some modeling guidelines presented in Section 12.5, to a form that is suitable for the isolability analysis that will be presented. The resulting model is shown in Table 12.1. For this example, we choose to consider derivatives as separate independent variables to illustrate this structural representation. When doing this, differentiation of model equations can be needed according to the discussion in Section 8.5. We will use a differentiated version of the model in Table 12.1 that is shown in Table 12.2. The order of differentiations of each equation is determined by an algorithm described in (Krysander and Nyberg,

Table 12.1: A reformulation of the model in Table 2.2

Assumption	Equation	Expression
Pump		
$P = \text{NF}$	e_1	$u = q_1$
Tank		
	e_2	$\dot{w} = q_1 - q_2$
	e_3	$w = (1 - f_c)q_2^2$
$T = \text{NF}$	e_4	$f_c = 0$
	$\dot{e}_4$	$\dot{f}_c = 0$
Water-level sensor		
$W = \text{NF}$	e_5	$y_w = w$
Flow sensor		
	e_6	$y_q = q_2 + f_{yq}$
$Q = \text{NF}$	e_7	$f_{yq} = 0$
	$\dot{e}_7$	$\dot{f}_{yq} = 0$

2002a) and (Krysander, 2003) such that all consistency relations including u, y_w, $\dot{y}_w$, y_q, and $\dot{y}_q$ can be derived from the differentiated model. The order of derivatives to include are inputs to the algorithm.

For the model in Table 12.2, the bias fault mode B of the outflow sensor Q can violate e_7 in the no-fault system behavioral-model $M_{\mathbf{NF}}$, because for example the constant signal $f_{yq} \equiv 1$ is consistent with the behavioral model $M_{\mathbf{Q}}$ but not with e_7. This means that the fault mode **Q** might be detectable.

Even if a fault can violate an equation in a model, it is not sure that the fault is detectable as the next small illustrative example shows.

Example 12.2
Consider a no-fault behavioral model $M_{\mathbf{NF}}$ defined as

Equation	Expression
e_1	$u = x_1$
e_2	$y = x_1$
e_3	$0 = x_1 + x_2$

(12.3)

where u and y are known variables and x_1 and x_2 are unknowns. The observation set of $M_{\mathbf{NF}}$ is

$$\mathcal{O}(M_{\mathbf{NF}}) = \{(u, y) \in \mathbb{R}^2 | u = y\} \tag{12.4}$$

A fault violating either e_1 or e_2 is detectable, because $u \neq y$ if either e_1 or e_2 is violated, i.e., (u, y) does not belong to the observation set (12.4). A fault which only violates e_3 cannot be detected because a violation of e_3 leads to different values of x_2 but $u = y$ still holds. A difference between the first two equations where a fault can be detected and e_3 where a fault can not be detected is that the first two equations define the observation set (12.4) and e_3 is not needed to define (12.4). This is in the linear case equal to that $M_{\mathbf{NF}}^* = \{e_1, e_2\}$ is the PO part of $M_{\mathbf{NF}} = \{e_1, e_2, e_3\}$.

As in (Blanke et al., 2003) we say that e_3 is *non-monitorable*. The concept of monitorable will be generalized from the definition in (Blanke et al., 2003) to models containing fault models as follows:

Table 12.2: A differentiated version of the model in Table 12.1.

Assumption	Equation	Expression
Pump		
$P = \text{NF}$	e_1	$u = q_1$
Tank		
	e_2	$\dot{w} = q_1 - q_2$
	e_3	$w = (1 - f_c)q_2^2$
	$\dot{e}_3$	$\dot{w} = 2\,(1 - f_c)\,q_2\dot{q}_2 - \dot{f}_c\,q_2^2$
$T = \text{NF}$	e_4	$f_c = 0$
	$\dot{e}_4$	$\dot{f}_c = 0$
Water-level sensor		
$W = \text{NF}$	e_5	$y_w = w$
$W = \text{NF}$	$\dot{e}_5$	$\dot{y}_w = \dot{w}$
Flow sensor		
	e_6	$y_q = q_2 + f_{yq}$
	$\dot{e}_6$	$\dot{y}_q = \dot{q}_2 + \dot{f}_{yq}$
$Q = \text{NF}$	e_7	$f_{yq} = 0$
	$\dot{e}_7$	$\dot{f}_{yq} = 0$

Definition 12.5 (Monitorable). *Given a set of equations M, an equation $e \in M$ is **monitorable** in M if there exists a model $M' \subseteq M$ such that*

$$\mathcal{O}(M') \subset \mathcal{O}(M' \setminus \{e\}) \tag{12.5}$$

An equation $e \in M$ that is not monitorable in M is said to be non-monitorable in M and fulfills

$$\mathcal{O}(M') = \mathcal{O}(M' \setminus \{e\}) \tag{12.6}$$

for all $M' \subseteq M$. For the model (12.3), equation e_1 and e_2 are monitorable equations and e_3 is non-monitorable. For a linear model M expressed as $H(p)\boldsymbol{x} + L(p)\boldsymbol{z} = 0$ where $[H\,L]$ has full row-rank an equation $e \in M$ is monitorable if and only if e belongs to the PO part. The following definition will be used to formalize in which equations violations can be detected.

Definition 12.6 (Checking Model of a Behavioral Mode). *A model C_b is a* checking model of *b if C_b is a subset of the behavioral model M_b and*

$$\mathcal{O}(C_b) = \mathcal{O}(M_b) \tag{12.7}$$

Note that behavioral models trivially are checking models. As examples of checking models, the two checking models of **NF** in (12.3) are the sets $\{e_1, e_2\}$ and $\{e_1, e_2, e_3\}$. Note also that checking models do not need to have more equations than unknowns. For example consider the behavioral model $\{y = x^2\}$ where y is assumed to be known and x is assumed to be unknown. This model is an example of such checking model.

A detectable fault violates at least one equation in every checking model $C_{\mathbf{NF}}$ for the no-fault behavioral mode. This means for the model in (12.3) that a detectable fault must violate e_1 or e_2, because $\{e_1, e_2\}$ is a checking model of **NF**.

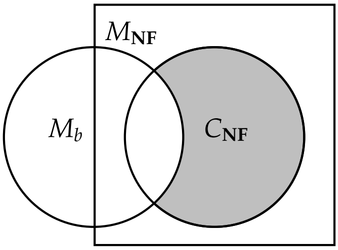

Figure 12.1: Venn-diagram representation of equation sets.

An example of the equation sets $M_{\mathbf{NF}}$, $C_{\mathbf{NF}}$, and M_b involved in the discussion about detectability of b is shown in Figure 12.1 as a Venn diagram. The rectangle represents the set of all equations in the no-fault behavioral model $M_{\mathbf{NF}}$, i.e., $\{e_1, e_2, e_3\}$ in the small example. The right circle contains a checking model $C_{\mathbf{NF}}$ of the no-fault behavioral mode, e.g., $\{e_1, e_2\}$ in the example. The left circle contains the behavioral model M_b for some behavioral mode b. The equations in $M_{\mathbf{NF}} \setminus M_b$ are the equations that might be violated by b. However, among these equations, it is only the equations in the grey-shaded area that render detection of behavioral mode b possible. Hence if the grey-shaded area is empty, then b is not detectable.

Example 12.3
Consider the model $M_{\mathbf{NF}} = \{e_1, e_2, e_3\}$ in (12.3). If $M_b = \{e_1, e_2\}$, then b is not detectable, because both e_1 and e_2 hold in b.

The next theorem formalizes the result of the discussion, i.e., how checking models will be used for detectability analysis.

Theorem 12.1. *Given a diagnosis model $\mathbb{M}$, a system behavioral-mode b is not detectable in $\mathbb{M}$, i.e., $(b, \mathbf{NF}) \notin I(\mathbb{M})$, if there exists a checking model $C_{\mathbf{NF}}$ of* **NF** *such that $C_{\mathbf{NF}} \subseteq M_b$.*

To prove this theorem, the following lemma will be used.

Lemma 12.1. *Given a diagnosis model $\mathbb{M}$, a system behavioral-mode b_i is not isolable from a system behavioral mode b_j in $\mathbb{M}$, i.e., $(b_i, b_j) \notin I(\mathbb{M})$, if and only if*

$$\mathcal{O}(M_{b_i}) \subseteq \mathcal{O}(M_{b_j}) \tag{12.8}$$

Proof. The mode b_i is not isolable from b_j if and only if whenever b_i is a diagnosis b_j is a diagnosis too. This can according to the definition of diagnosis be written as $z \in \mathcal{O}(M_{b_i})$ implies that $z \in \mathcal{O}(M_{b_j})$. This implication is equivalent to (12.8) which completes the proof. □

Now, we are ready to prove Theorem 12.1.

Proof. From $C_{\mathbf{NF}} \subseteq M_b$ it follows that

$$\mathcal{O}(M_b) \subseteq \mathcal{O}(C_{\mathbf{NF}})$$

This and Definition 12.6 imply that

$$\mathcal{O}(M_b) \subseteq \mathcal{O}(M_{\mathbf{NF}}) \tag{12.9}$$

since $C_{\mathbf{NF}}$ is a checking model of **NF**. Equation (12.9) and Lemma 12.1 imply that b is not isolable from **NF** which means that b is not detectable. □

By finding a checking model of behavioral mode **NF** such that $C_{\mathbf{NF}} \subseteq M_b$, it follows from Theorem 12.1 that b is not detectable. How to find checking models will be described in Section 12.2 and Section 12.3.

Example 12.4
For the water-tank process with the model in Table 2.2, all equations in the no-fault behavioral model are needed to define the observation set for the no-fault behavioral-model. This is not trivial to determine but true, i.e., the checking model for no-fault is equal to the no-fault behavioral model. It holds that

$$C_{\mathbf{NF}} \not\subseteq M_b \tag{12.10}$$

for any single fault behavioral mode b and this implies according to Theorem 12.1 that no single fault can be said to be non-detectable, i.e., all single faults might be detectable.

12.1.2 Predicting Isolability

Since detectability is a special case of isolability, the results of Theorem 12.1 concerning detectability can be generalized to isolability as follows. A behavioral mode b_i, that is isolable from a behavioral mode b_j, violates some equations in a checking model C_{b_j} of the behavioral mode b_j. Figure 12.1 could represent this situation as well if **NF** is changed to b_j and b to b_i. Then it can be seen that if all equations in a checking model C_{b_j} hold in behavioral mode b_i then it follows that b_i is not isolable from b_j. Hence by computing a checking model of C_{b_j}, it can be concluded which behavioral modes that are not isolable from b_j.

Theorem 12.2. *Given a diagnosis model $\mathbb{M}$, a system behavioral-mode b_i is not isolable from a system behavioral mode b_j in $\mathbb{M}$, i.e., $(b_i, b_j) \notin I(\mathbb{M})$, if there exists a checking model C_{b_j} of b_j such that*

$$C_{b_j} \subseteq M_{b_i} \tag{12.11}$$

Proof. Inclusion (12.11) implies that

$$\mathcal{O}(M_{b_i}) \subseteq \mathcal{O}(C_{b_j})$$

which according to Definition 12.6 implies that

$$\mathcal{O}(M_{b_i}) \subseteq \mathcal{O}(M_{b_j})$$

Hence b_i is not isolable from b_j according to Lemma 12.1. □

In conclusion, by computing a checking model for each system behavioral-mode, Theorem 12.1 and Theorem 12.2 give an explicit method to compute if a faulty behavioral mode is not detectable and if a behavioral mode is not isolable from another behavioral mode. To summarize the isolability prediction, i.e., which behavioral modes that might be isolable from others and which faults that might be detectable in a diagnosis model according to the set of checking models, the following notation will be used. Given a diagnosis model $\mathbb{M}$, and a checking model $C_{b_i} \subseteq M_{b_i}$ for each behavioral mode $b_i \in \mathbf{B}$, a binary relation $I_p(\langle C_{b_1}, \ldots, C_{b_n} \rangle)$ on $\mathbf{B} \times \mathbf{B}$ is an isolability prediction of $I(\mathbb{M})$ if

$$I_p(\langle C_{b_1}, \ldots, C_{b_n} \rangle) = \{(b_i, b_j) \in \mathbf{B} \times \mathbf{B} | C_{b_j} \not\subseteq M_{b_i}\} \tag{12.12}$$

This expression can be used for computing the isolability prediction given checking models for all behavioral modes. From Theorem 12.2, the following result about the relation between an isolability prediction of $I(\mathbb{M})$ and the isolability of a diagnosis model $I(\mathbb{M})$ is immediate.

Corollary 12.1. *If $\mathbb{M}$ is a diagnosis model and $C_{b_i} \subseteq M_{b_i}$ is checking model for each behavioral mode $b_i \in \mathbf{B}$, then*

$$I(\mathbb{M}) \subseteq I_p(\langle C_{b_1}, \ldots, C_{b_n} \rangle) \tag{12.13}$$

Proof. An equivalent formulation of (12.13) is

$$I_p(\langle C_{b_1}, \ldots, C_{b_n} \rangle)^C \subseteq I(\mathbb{M})^C \tag{12.14}$$

For an arbitrary pair $(b_i, b_j) \in I_p(\langle C_{b_1}, \ldots, C_{b_n} \rangle)^C$, it holds according to (12.12) that $C_{b_j} \subseteq M_{b_i}$. This and Theorem 12.2 implies that $(b_i, b_j) \in I(\mathbb{M})^C$, which completes the proof. □

This corollary states that the isolability prediction is an upper bound for the isolability of the diagnosis model.

12.1.3 Isolability and Checking Models

There might exist several checking models of a system behavioral-mode b_j as seen previously, and depending on which checking models that are used different isolability predictions $I_p(\langle C_{b_1}, \ldots, C_{b_n} \rangle)$ may be obtained. In this section, we investigate which checking models that gives a least upper bound $I_p(\langle C_{b_1}, \ldots, C_{b_n} \rangle)$ of $I(\mathbb{M})$. Assume that one checking model $C^1_{b_j}$ is a proper subset of another checking model $C^2_{b_j}$, i.e.,

$$C^1_{b_j} \subset C^2_{b_j}$$

If it holds that

$$C^2_{b_j} \subseteq M_{b_i}$$

then

$$C^1_{b_j} \subseteq M_{b_i}$$

but the opposite does not hold. This and Theorem 12.2 imply that if checking model $C^2_{b_j}$ implies that b_i is not isolable from b_j then $C^1_{b_j}$ does that too. Now assume that

$$C^1_{b_j} \subset M_{b_i} \subset C^2_{b_j}$$

By using $C^1_{b_j}$ as checking model for b_j, it is concluded from Theorem 12.2 that b_i is not isolable from b_j. However if $C^2_{b_j}$ is used as checking model then no conclusion can be drawn. Hence the strongest conclusion is given by the minimal checking models. There might exist several minimal checking models of a behavioral mode in the general case. For example a linear model with linearly dependent equations may have several minimal checking models. However, the methods, that we later propose, for computing checking models will output a single checking model of each behavioral mode. Therefore, we will for convenience only discuss the case when using one checking model of each behavioral mode. The extension to use several minimal checking models of a behavioral mode can be done as follows. If $C^1_{b_1}$ and $C^2_{b_1}$ are two checking models of b_1, then an isolability prediction based on both these models is

$$I_p(\langle C^1_{b_1}, C_{b_2}, \ldots, C_{b_n}\rangle) \cap I_p(\langle C^2_{b_1}, C_{b_2}, \ldots, C_{b_n}\rangle) \tag{12.15}$$

In conclusion, by finding smaller checking models than M_b more faults can be concluded not to be isolable from others, i.e., the isolability prediction becomes better. In general, for two different choices of checking models

$$\langle C^1_{b_1}, \ldots, C^1_{b_n}\rangle \tag{12.16}$$

and

$$\langle C^2_{b_1}, \ldots, C^2_{b_n}\rangle \tag{12.17}$$

where

$$C^1_{b_i} \subseteq C^2_{b_i} \tag{12.18}$$

for all behavioral modes $b_i \in \mathbf{B}$, it follows that

$$I(\mathbb{M}) \subseteq I_p(\langle C^1_{b_1}, \ldots, C^1_{b_n}\rangle) \subseteq I_p(\langle C^2_{b_1}, \ldots, C^2_{b_n}\rangle) \tag{12.19}$$

The minimal checking models of a system behavioral-mode are unknown and depends on the analytical expressions of the equations in the model. A brute-force approach to compute the minimal checking models would be to compute observation sets for subsets of equations and compare it to the observation set of the behavioral model. Even for models of the size and complexity like the water-tank example, automatic computation of observation sets by using computer algebra, like for example Mathematica or Maple, is computationally demanding. For a large industrial example this approach would be computationally intractable. In the next two sections, two different structural methods will be presented to compute the smallest checking model of b.

12.2 Finding Checking Models by Using PSO Parts

The first method will be to use the PSO part of M_b as checking model of b. To motivate this approach, a linear model is first considered.

Theorem 12.3. *Given a structured linear behavioral model M_b, it follows that M_b^+ is the only minimal checking model of b.*

Proof. Let the set M_b of equations be written as

$$\begin{bmatrix} H_{11}(p) & H_{12}(p) \\ 0 & H_{22}(p) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} L_1(p) \\ L_2(p) \end{bmatrix} z = 0 \tag{12.20}$$

where the first row corresponds to $(M_b \setminus M_b^+)$ and the second row to the equations M_b^+. From the definition of Dulmage-Mendelsohns decomposition in Section 9.1.2, it follows that $H_{11}(p)$ has full structural row rank. Since the model is structured, the matrix $H_{11}(p)$ is a structured matrix. This implies that $H_{11}(p)$ has full row-rank. Since $H_{11}(p)$ has full row rank, it follows that $N_H[:, M_b \setminus M_b^+] = 0$. Then we get that

$$\mathcal{O}(M_b^+) = \{z \in \mathcal{Z} | N_{H_{22}}(p)L_2(p)z = 0\} = \{z \in \mathcal{Z} | N_H(p)L(p)z = 0\} = \mathcal{O}(M_b) \quad (12.21)$$

Hence M_b^+ is a checking model of b.

Now, we will prove that M_b^+ is a minimal checking model. Since M_b^+ is a PSO set and these equations are structured, it follows from Theorem 8.6 that M_b^+ is a PO set. By definition, this implies that $\varphi(E) < \varphi(M_b^+)$ for any proper subset $E \subset M_b^+$. This means that $\mathcal{O}(M_b^+) \subset \mathcal{O}(E)$ and by using also that (12.21) it follows that E is not a checking model of b. Hence M_b^+ is a minimal checking model.

Finally, we address the uniqueness. Let M be an arbitrary minimal checking model of b. Since H_{11} has full row-rank, it follows that $M \cap (M_b \setminus M_b^+) = \emptyset$ or equivalently that $M \subseteq M_b^+$. From the minimality of M_b^+, it follows that $M = M_b^+$. Since M was an arbitrary minimal checking model, it follows that M_b^+ is the only minimal checking model of b and this completes the proof. □

As a consequence of Theorem 12.3, we can predict isolability according to the following corollary.

Corollary 12.2. *Given a diagnosis model $\mathbb{M}$ where the model equations $\boldsymbol{M}$ are structured and linear, it follows that b_i is not isolable from b_j in $\mathbb{M}$ if*

$$M_{b_j}^+ \subseteq M_{b_i} \quad (12.22)$$

Proof. This corollary follows directly from Theorem 12.2 and Theorem 12.3 □

Theorem 12.3 states that the minimal checking model of a behavioral mode b is for a generic linear model M_b equal to the PSO part of M_b. Therefore it is reasonable to use the PSO part of the behavioral model as the checking model, i.e., $C_b = M_b^+$. Next, we will show for linear structured systems under just slightly stronger conditions than in Theorem 12.3 that the isolability prediction is exactly equal to the true isolability. Since it is not possible to derive that a behavioral mode is isolable from another behavioral mode by using Theorem 12.2, we will show this by using other methods in the next lemma and theorem.

Lemma 12.2. *Let M be a set of linear dynamic equations*

$$H(p)x + L(p)z = 0 \quad (12.23)$$

such that $[H\,L]$ has full row-rank. If $M_1 \subseteq M$ is a PO set and $M_2 \subseteq M$ is an arbitrary set, it follows that

$$M_1 \subseteq M_2 \quad (12.24)$$

if and only if

$$\mathcal{O}(M_2) \subseteq \mathcal{O}(M_1) \quad (12.25)$$

Proof. The only if-direction is trivially true. Hence it remains to prove the if-direction. Assume that (12.25) is true. This is according to Theorem 6.2 equivalent to that

$$N_2 L z = 0 \quad (12.26)$$

implies

$$N_1 L z = 0 \tag{12.27}$$

where $N_i = N_{H[M_i]}$. This is true if and only if there exists a matrix U of suitable dimension such that

$$U N_2 L = N_1 L \tag{12.28}$$

Since N_1 and N_2 belong to the left null space of H, it follows that (12.28) is equivalent to

$$(U N_2 - N_1)[H\, L] = 0 \tag{12.29}$$

Since $[H\, L]$ has full row-rank, we get equivalence between (12.29) and

$$U N_2 = N_1 \tag{12.30}$$

From the definition of N_2, it follows that $N_2[:, M \setminus M_2] = 0$. This and (12.30) imply that

$$N_1[:, M \setminus M_2] = U N_2[:, M \setminus M_2] = 0 \tag{12.31}$$

Since M_1 is a PO set, it follows from Corollary 3.1, which is true also for the dynamic case, that M_1 is the set non-zero columns in N_1. This and $N_1[:, M \setminus M_2] = 0$, imply that $M_1 \cap (M \setminus M_2) = \emptyset$, or equivalently that (12.24) holds. □

In the following theorem, we will used the notation introduced in Section 8.3 of a bipartite graph $G(\boldsymbol{M}, \boldsymbol{X} \cup \boldsymbol{Z})$ with equations M and variables $\boldsymbol{X} \cup \boldsymbol{Z}$ as node sets.

Theorem 12.4. *Let $\mathbb{M}$ be a diagnosis model where the model equations $\boldsymbol{M}$ on the form*

$$H(p)\boldsymbol{x} + L(p)\boldsymbol{z} = 0 \tag{12.32}$$

are structured and $G(\boldsymbol{M}, \boldsymbol{X} \cup \boldsymbol{Z})$ has a complete matching of $\boldsymbol{M}$ into $\boldsymbol{X} \cup \boldsymbol{Z}$. Then it holds that b_i is isolable from b_j in $\mathbb{M}$ if and only if

$$M^+_{b_j} \not\subseteq M_{b_i} \tag{12.33}$$

Proof. Since $\boldsymbol{M}$ is a structured model, it follows from Theorem 12.3 that $M^+_{b_j}$ is a PO set and a minimal checking model of b_j, i.e., $\mathcal{O}(M^+_{b_i}) = \mathcal{O}(M_{b_i})$. Furthermore, since $G(\boldsymbol{M}, \boldsymbol{X} \cup \boldsymbol{Z})$ has a complete matching of $\boldsymbol{M}$ into $\boldsymbol{X} \cup \boldsymbol{Z}$ and $\boldsymbol{M}$ is structured, it follows that $[H\, L]$ has full row rank. This means that the condition in Lemma 12.2 is fulfilled. From Lemma 12.2 and the fact that $M^+_{b_j}$ is a PO set, it follows that (12.33) is equivalent to

$$\mathcal{O}(M_{b_i}) \not\subseteq \mathcal{O}(M^+_{b_j}) = \mathcal{O}(M_{b_j}) \tag{12.34}$$

This is by definition equivalent to that b_i is isolable from b_j and the theorem follows. □

Note that M_{b_i} in (12.33) can be replaced by $M^+_{b_i}$. This theorem will next be illustrated by an example. Before we exemplify this, note that a *diagnosis implication partial order* $P(\mathbb{M})$ can be defined based on $I(\mathbb{M})$ in a similar way to the partial orders defined in Definition 11.5.

Example 12.5

Consider the electrical circuit in Example 3.17. In Figure 3.1, all PO subsets of the model (3.95) are shown. The PSO sets are exactly equal to the PO sets in this example. For each behavioral mode b, the behavioral mode is written directly below its checking model defined as $C_b = M_b^+$ in Figure 3.1. If this Hasse-diagram is flipped horizontally, the obtained Hasse-diagram shows the diagnosis implication partial order that corresponds to the isolability of the diagnosis model. The equivalence classes are the sets of behavioral modes that correspond to each PSO set. One example of an equivalence class is {**R1**,**S2**,**R1**&**S2**} and these behavioral modes are not isolable from each other. Another interpretation of the partial order is that **NF** is isolable from exactly all behavioral modes where the battery is faulty.

Hence, this example shows that the isolability of the diagnosis model can exactly be computed by using only structural methods.

The method to use the PSO part of a behavioral model as the checking model can be applied also for non-linear models. The PSO part M_b^+ is easily computed for any model M_b. However, a disadvantage by using the PSO part as the checking model is that there exist behavioral models M_{b_j} where $M_{b_j}^+$ is not a checking model of b_j. If this is the case and $M_{b_j}^+ \subseteq M_{b_i}$, then b_i might be isolable from b_j because the condition of Theorem 12.2 is not fulfilled. A non-linear example of this will be given later in Example 12.12. In (Blanke et al., 2003; Frisk et al., 2003; Pulido and Alonso, 2002) analytical assumptions imply that M_b^+ is a checking model of b, but the assumptions may be difficult to verify.

12.3 Finding Checking Models by Removing Equations

A problem with the first approach for computing checking models is that it can be difficult to verify that M_b^+ is a checking model of b. In this section a second approach for finding checking models will be presented. An advantage with this approach is that no difficult analytical assumptions need to be fulfilled to find checking models. To do this, we propose to compute the smallest checking model of b, that can be obtained with an alternative structural method to be presented in Section 12.3.3 and not requiring an exact determination of all minimal checking models of b. The checking model of b obtained with the structural method to be presented in Section 12.3.3 will in the continuation be called the smallest checking model for b. The strategy to find the smallest checking model of b will be to start with the corresponding behavioral model and remove equations which are not needed to define the observation set for the behavioral model, i.e., to remove non-monitorable equations.

12.3.1 Excluding Non-monitorable Equations

Remember that if X is any set of variables, then $\mathbf{x}$ will denote the vector of the variables in X. Consider a set of equations M with unknown variables $\boldsymbol{X}$ and known variables $\boldsymbol{Z}$. Let $\boldsymbol{X}$ be partitioned into X_1 and X_2 and let the domain of $\mathbf{x}_i$ be denoted by $\mathbb{X}_i$. If the set M fulfills

$$\forall \mathbf{z} \in \mathbb{Z} \forall \mathbf{x}_2 \in \mathbb{X}_2 \exists \mathbf{x}_1 \in \mathbb{X}_1 : M(\mathbf{x}_1, \mathbf{x}_2, \mathbf{z}) \tag{12.35}$$

then the set M of equations is said to be X_1-*satisfiable*. Next an example of this notation is given.

Example 12.6
Consider the model in Table 12.2. For example, let $M = \{e_3\}$ and $X_1 = \{w\}$. For arbitrary values of f_c and q_2 there exists a value $w = (1 - f_c)q_2^2$ such that e_3 is true, i.e., $\{e_3\}$ is $\{w\}$-satisfiable.

The following theorem describes how the satisfiable notion can be used to exclude non-monitorable equations from checking models.

Theorem 12.5. *If a model $M \subseteq M_b$ is X_1-satisfiable and no variable in X_1 is contained in $M_b \backslash M$, then $M_b \backslash M$ is a checking model of b.*

Proof. Let M_b be a behavioral model with Z as the set of its known variables and with $X_1 \cup X_2$ as a partition of the set of its unknowns. Let $\bar{M} = M_b \backslash M$. Since the variables in X_1 are not included in $\bar{M}$ the observation set $\mathcal{O}(M_b)$ is

$$\mathcal{O}(M_b) = \{z \in \mathbb{Z} | \exists x_1 \in \mathbb{X}_1, x_2 \in \mathbb{X}_2 : (\bar{M}(x_1, z) \wedge M(x_1, x_2, z))\} \quad (12.36)$$

The set in (12.36) can be expressed as

$$\{z \in \mathbb{Z} | \exists x_2 \in \mathbb{X}_2 : (\bar{M}(x_2, z) \wedge \exists x_1 \in \mathbb{X}_1 : M(x_1, x_2, z))\} = \{z \in \mathbb{Z} | \exists x_2 \in \mathbb{X}_2 : \bar{M}(x_2, z)\} \quad (12.37)$$

where the equality holds since M is X_1-satisfiable. The last set is equal to $\mathcal{O}(\bar{M})$ which implies that

$$\mathcal{O}(M_b) = \mathcal{O}(\bar{M}) \quad (12.38)$$

This and Definition 12.6 implies that $\bar{M} = M_b \setminus M$ is a checking model for b which was to be proved. □

An alternative formulation of Theorem 12.5 is that if M is X_1-*satisfiable* and no variable in X_1 is contained in $M_b \backslash M$, then M is non-monitorable. This means that a checking model smaller than the behavioral model can be computed by removing equation set M from the behavioral model M_b.

Example 12.7
To give an example of how this is done, consider the behavioral mode **W** for the water-tank example. Since $\{e_3\}$ is $\{w\}$-satisfiable and e_3 is the only equation in $M_{\mathbf{W}}$ where w is included, $M_{\mathbf{W}} \backslash \{e_3\}$ is a checking model of **W** according to Theorem 12.5. Hence, a smaller checking model than $M_{\mathbf{W}}$ has been identified.

This example shows that structural information together with satisfiability information is sufficient for identifying non-monitorable equations, and then a smaller checking model can be identified.

12.3.2 Structural Method

A structural method will be used for determining satisfiability. In fact, non-monitorable equation sets in a behavioral model can be identified by using only structural information. The structure that will be used as input to the method includes additional information compared to the structural representations described in Section 8.5, and this will be explained by an example. The structure of the behavioral mode $M_{\mathbf{PW}}$ in the water-tank model in Table 12.2 is represented as a biadjacency matrix in Table 12.3. An "X" or an "O" in row e and column x means, as before, that x is included in e. An entry corresponding to equation e and variable x is marked "X" if $\{e\}$ is $\{x\}$-satisfiable and otherwise "O". Insights of the physics can be used to specify where to put "X":s.

By using this additional information in the structure, it is possible to find non-monitorable equation sets with cardinality one as follows. If e is the only equation in M_b that contains a variable x and this variable is marked with an "X" in the biadjacency matrix, then $\{e\}$ satisfies the conditions in Theorem 12.5, i.e., $\{e\}$ is non-monitorable. The next theorem will give theoretical results needed for computing non-monitorable equation sets with cardinality greater than 1.

Theorem 12.6. *Let M_1 and M_2 be disjoint sets of equations. If M_1 is X_1-satisfiable, M_2 is X_2-satisfiable and does not contain any variable in X_1, then it follows that $M_1 \cup M_2$ is $(X_1 \cup X_2)$-satisfiable.*

Proof. Let the set of variables in $M_1 \cup M_2$ not included in either X_1 or X_2 be denoted X_3. From the conditions on M_1 and M_2, it follows that

$$\forall \boldsymbol{x}_3 \in \mathbb{X}_3 \forall \boldsymbol{x}_2 \in \mathbb{X}_2 \exists \boldsymbol{x}_1 \in \mathbb{X}_1 : M_1(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) \wedge \forall \boldsymbol{x}_3 \in \mathbb{X}_3 \exists \boldsymbol{x}_2 \in \mathbb{X}_2 : M_2(\boldsymbol{x}_2, \boldsymbol{x}_3)$$

which implies that

$$\forall \boldsymbol{x}_3 \in \mathbb{X}_3 \exists (\boldsymbol{x}_1, \boldsymbol{x}_2) \in \mathbb{X}_1 \times \mathbb{X}_2 : (M_1(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) \wedge M_2(\boldsymbol{x}_2, \boldsymbol{x}_3))$$

since for any x_3 there exists an x_2 consistent with M_2 and for any x_2 there exists an x_1 consistent with M_1 and therefore also M_1 and M_2. This and the definition of X-satisfiable models imply that $M_1 \cup M_2$ is $X_1 \cup X_2$-satisfiable. □

Example 12.8

To exemplify how the result of Theorem 12.6 can be used, consider the behavioral model $M_b = M_{\mathbf{PW}}$ in the water-tank example. The structure of the model $M_{\mathbf{PW}}$ is shown in Table 12.3. The model $M_1 = \{e_2\}$ is $\{q_1\}$-satisfiable and $M_2 = \{\dot{e}_3\}$ is $\{\dot{w}\}$-satisfiable. Now, since $\{\dot{e}_3\}$ and $\{e_2\}$ are disjoint and q_1 is not included in $\dot{e}_3$, Theorem 12.6 implies that $\{e_2, \dot{e}_3\}$ is $\{q_1, \dot{w}\}$-satisfiable. Furthermore, the variables in $\{q_1, \dot{w}\}$ are not included in $M_{\mathbf{PW}} \setminus \{e_2, \dot{e}_3\}$ which means that $M_{\mathbf{PW}} \setminus \{e_2, \dot{e}_3\}$ is a checking model of **PW**, according to Theorem 12.5.

In this way, it is possible to find the smallest checking model by finding a non-monitorable equation and remove them from the model.

12.3.3 Algorithm

In this section we will present a recursive algorithm for computing the smallest possible checking model of a behavioral mode b given a structural model of the type shown in Table 12.3. The input to the algorithm is a structure with "O":s and "X":s of a behavioral model $M_b =: C_b$.

Table 12.3: The structure of the behavioral model $M_{\mathbf{PW}}$ included in the diagnosis model shown in Table 12.2.

Equation	Unknowns									Knowns				
	q_1	w	$\dot{w}$	q_2	$\dot{q}_2$	f_c	$\dot{f}_c$	f_q	$\dot{f}_q$	u	y_w	$\dot{y}_w$	y_q	$\dot{y}_q$
e_2	X		X	X										
e_3		X		O		O								
$\dot{e}_3$			X	O	O	O	O							
e_4						X								
$\dot{e}_4$							X							
e_6				X				X					X	
$\dot{e}_6$					X				X					X
e_7								X						
$\dot{e}_7$									X					

Algorithm 13. `FindCheckingModel`(C_b)

```
if $C_b$ contains no known variable then
    $C_b := \emptyset$
else if there exists an $e \in C_b$ with an unknown $x$ only in $e$ and the entry $(e, x)$ is marked "X" then
    $C_b :=$ FindCheckingModel($C_b \backslash \{e\}$);
end if
return The checking model $C_b$.
```

The correctness of the algorithm is implied by Theorem 12.5 and Theorem 12.6. For a checking model C_b obtained by Algorithm 13, it holds that $M_b^+ \subseteq C_b \subseteq M_b$. Note that the checking model computed by Algorithm 13 contains all algebraic loops contained in the behavioral model M_b. No algebraic loop can be removed by using only the information available in the input to Algorithm 13 and this will be illustrated next by an example.

Example 12.9
Consider a behavioral model

Equation	Expression
e_1	$y_1 = x_1 + x_2$
e_2	$y_2 = x_1 + x_2$

(12.39)

and another behavioral model

Equation	Expression
e_1	$y_1 = x_1 + x_2$
e_2	$y_2 = x_1 + 2x_2$

(12.40)

For the model in (12.39), the minimal checking model is $\{e_1, e_2\}$ and for the model in (12.40) the empty set is the minimal checking model. Both these two behavioral models have the same structure:

Equation	Unknowns		Knowns	
	x_1	x_2	y_1	y_2
e_1	X	X	X	
e_2	X	X		X

(12.41)

From this structure, it is not possible to conclude which equations that are monitorable, because we have seen examples of behavioral models with exactly this structure, where all and none of the equations are monitorable. Hence, the only subset of $\{e_1, e_2\}$ that can be determined to be a checking model by only using the structural model (12.41) is $\{e_1, e_2\}$.

This was a linear example, but the same arguments hold also for non-linear models. If we in the example know that $\{e_1, e_2\}$ is $\{x_1, x_2\}$-satisfiable, then the empty checking model can be computed by using Theorem 12.5 and Theorem 12.6 directly. In this case, the algebraic loop would be excluded from the checking model. However, it is in general a difficult task to provide satisfiability information for different equation sets and variables sets of cardinalities greater than one. Therefore, it is reasonable to assume that only a structure of the type in Table 12.3 is provided.

The following example illustrates how Algorithm 13 is applied to a model.

Example 12.10
Consider for the water-tank example, the behavioral model $M_{\mathbf{PW}} = \boldsymbol{M}\setminus\{e_1, e_5, \dot{e}_5\}$ in the diagnosis model shown in Table 12.2. The structure of this model is seen in Table 12.3. `FindCheckingModel` is first called with input $M_b = M_{\mathbf{PW}}$. The variable q_1 is, among the equations in $M_{\mathbf{PW}}$, only included in e_2 and the corresponding entry is marked "X", i.e., the if-condition is satisfied and `FindCheckingModel` is called with input $M_{\mathbf{PW}}\setminus\{e_2\}$. Now the if-condition is also satisfied, because w is only included in e_3 and (e_3, w) is marked "X". By continuing the recursion in this way, `FindCheckingModel`$(M_{\mathbf{PW}})$ returns the empty set $\emptyset$ which is the checking model of **PW** to be used in the isolability computation later. This means according to Theorem 12.2, that no behavioral mode is isolable from that **PW**. In other words, **PW** is always a diagnosis.

12.4 Computation of Isolability Prediction

In the previous section, two different ways of computing checking models have been presented. Given checking models for all behavioral modes, it is straightforward to compute an isolability prediction using (12.12). However, this might be computationally inefficient, because the number n of behavioral modes can be large and (12.12) requires the determination of n^2 subset-relations.

In this section, we will propose more efficient methods to compute the isolability prediction $I_p(\langle C_{b_1}, \ldots, C_{b_n}\rangle)$. These methods will be based on the structural isolability defined in Definition 11.4. Next, we extend this definition to be applicable also to sets of models by using that $\Phi_i = \text{assump}(M_i)$.

Definition 12.7 (Structural Isolability of a Set of Models). *Let $\mathbb{M}$ be a diagnosis model and let ω be a set of models contained in $\mathbb{M}$. A binary relation $I_s(\omega)$ on $\mathbf{B} \times \mathbf{B}$ is* *the* ***structural isolability of a set of models*** ω *if*

$$I_s(\omega) = \{(b_1, b_2) | \exists M_i \in \omega : \Big(b_1 \notin \text{assump}(M_i) \wedge b_2 \in \text{assump}(M_i)\Big)\} \tag{12.42}$$

From the correspondence between the structural isolability of a diagnosis system and the structural isolability of a set of models, it follows from Theorem 11.2 that

$$I_s(\omega) = \bigcup_{M_i \in \omega} I_s(\{M_i\}) \tag{12.43}$$

This form is suitable for computing the structural isolability $I_s(\omega)$ and next we will show that the structural isolability of a set of models can be used to compute the proposed isolability predictions.

Consider the isolability prediction obtained by using the checking models suggested in Section 12.2, i.e., to use M_b^+ as checking models. Then the isolability prediction can be computed by using the following result.

Theorem 12.7. *Let* $\mathbb{M}$ *be a diagnosis model and let* ω_{MSO} *be the set of all feasible MSO sets contained in* $\mathbb{M}$*. Then, it follows that*

$$I_p(\langle M_{b_1}^+, \ldots, M_{b_n}^+ \rangle) = I_s(\omega_{MSO}) \tag{12.44}$$

Before we prove Theorem 12.7, a similar result for the approach presented in Section 12.3 is first stated. In $I_p(\langle C_{b_1}, \ldots, C_{b_n} \rangle)$ the checking models of all behavioral modes must be considered even if some or many of the checking models are equal to other checking models. The next theorem shows that identical checking models need not be considered separately.

Theorem 12.8. *Let* $\mathbb{M}$ *be a diagnosis model and let the checking models found by Algorithm 13 be* $\langle C_{b_1}, \ldots, C_{b_n} \rangle$*. If* $\omega_c = \{C_{b_i} | b_i \in \mathbf{B}\}$*, then it follows that*

$$I_p(\langle C_{b_1}, \ldots, C_{b_n} \rangle) = I_s(\omega_c) \tag{12.45}$$

Before we prove the theorems, the advantage of using the structural isolability to compute the isolability prediction is illustrated.

Example 12.11

Consider the electrical circuit example, it can be seen in Figure 2.3 that there are $2^5 = 32$ behavioral modes. If the isolability prediction is computed by using (12.12), then there are $32^2 = 1024$ set comparisons needs to be done. If the (12.43) is used, then the isolability prediction is computed directly from the isolability of the 8 MSO sets.

To prove the two theorems, the next lemma will be used. In the lemma we need to relate the set ω in the expression for structural isolability to the tuple of checking models $\langle C_{b_1}, \ldots, C_{b_n} \rangle$ in the isolability prediction. For that purpose the following notion will be used. Let $E_b(\omega)$ be the model $E_b \subseteq M_b$ such that

$$E_b(\omega) = \bigcup_{\substack{M \in \omega \\ M \subseteq M_b}} M \tag{12.46}$$

Then different sets ω implies different tuples $\langle E_{b_1}(\omega), \ldots, E_{b_n}(\omega) \rangle$.

We will later show that if $\omega = \omega_{MSO}$ in (12.46), then

$$E_b(\omega_{MSO}) = M_b^+ \tag{12.47}$$

for all $b \in \mathbf{B}$. Furthermore, we will also show that, if $\omega = \omega_c$ in (12.46), then

$$E_b(\omega_c) = C_b \tag{12.48}$$

for all $b \in \mathbf{B}$. Then both two theorems follow directly from the next lemma.

Lemma 12.3. *If* $\mathbb{M}$ *is a diagnosis model,* ω *is an arbitrary set of models, and* $E_b(\omega)$ *is defined as in (12.46), then*

$$I_s(\omega) = I_p(\langle E_{b_1}(\omega), \ldots, E_{b_n}(\omega)\rangle) \tag{12.49}$$

Proof. First, note that (12.42) is equivalent to

$$I_s(\omega) = \{(b_1, b_2)|\exists M_i \in \omega : \left(M_i \not\subseteq M_{b_1} \wedge M_i \subseteq M_{b_2}\right)\} \tag{12.50}$$

We start to show that

$$I_s(\omega) \subseteq I_p(\langle E_{b_1}(\omega), \ldots, E_{b_n}(\omega)\rangle) \tag{12.51}$$

Let (b_1, b_2) be arbitrarily chosen such that

$$(b_1, b_2) \in I_s(\omega) \tag{12.52}$$

holds. This is according to (12.50) equivalent to that there is a model $M \in \omega$ that fulfills

$$M \not\subseteq M_{b_1} \tag{12.53}$$

and

$$M \subseteq M_{b_2} \tag{12.54}$$

From (12.54), (12.46), and the fact that $M \in \omega$, we get that

$$M \subseteq E_{b_2}(\omega) \tag{12.55}$$

The expressions (12.53) and (12.55) imply

$$E_{b_2}(\omega) \not\subseteq M_{b_1} \tag{12.56}$$

This is according to (12.12) equivalent to

$$(b_1, b_2) \in I_p(\langle E_{b_1}(\omega), \ldots, E_{b_n}(\omega)\rangle) \tag{12.57}$$

Hence since $(b_1, b_2) \in I_s(\omega)$ was arbitrarily chosen, it follows that (12.51) is true.

Now, it remains to prove that

$$I_s(\omega) \supseteq I_p(\langle E_{b_1}(\omega), \ldots, E_{b_n}(\omega)\rangle) \tag{12.58}$$

Let (b_1, b_2) be arbitrarily chosen such that

$$(b_1, b_2) \in I_p(\langle E_{b_1}(\omega), \ldots, E_{b_n}(\omega)\rangle) \tag{12.59}$$

This is according to (12.12) equivalent to (12.56). This means according to (12.46) that there is a model $M \in \omega$ that fulfills (12.53) and (12.54). This is equivalent to (12.52) and since (b_1, b_2) is arbitrarily chosen such that (12.59) is fulfilled, (12.58) follows. The theorem follows from (12.51) and (12.58). □

Now the proof of Theorem 12.7 follows.

Proof. First, we show that (12.47) holds. Let E be an arbitrary MSO set such that $E \subseteq M_b$. By substituting $M \setminus \{e\}$ for M_b in Theorem 9.6, it follows that $E \subseteq M_b^+$. Since E was an arbitrarily chosen MSO set such that $E \subseteq M_b$, (12.46) implies that

$$E_b(\omega_{MSO}) \subseteq M_b^+ \tag{12.60}$$

Now, the opposite inclusion will be proved. If $M_b^+ = \emptyset$, then this inclusion is trivially true. Hence assume that $M_b^+ \neq \emptyset$. Let e' be an arbitrary equation in M_b^+. An MSO set containing e' can then be constructed, by applying the subroutine in Algorithm 11 to $\mathcal{S} = \{\{e\}|e \in M_b^+\}$ and $\mathcal{E} = \mathcal{S} \setminus \{\{e'\}\}$. The output of the algorithm contains at least on MSO sets $M \in \omega_{MSO}$ such that $M \subseteq M_b^+ \subseteq M_b$ and $e' \in M$. This and (12.46) imply that $e' \in M \subseteq E_b(\omega_{MSO})$. Since e' was an arbitrarily chosen equation in M_b^+, it follows that

$$E_b(\omega_{MSO}) \supseteq M_b^+ \tag{12.61}$$

From (12.60) and (12.61), we get that (12.47) is true. This and Lemma 12.3 imply (12.44) and this completes the proof. □

Now, the proof of Theorem 12.8 follows.

Proof. First, we show that (12.48) holds. Since $C_b \in \omega_c$ and $C_b \subseteq M_b$, it follows that

$$E_b(\omega_c) \supseteq C_b \tag{12.62}$$

Hence, it remains to prove the opposite inclusion. Let M_1 and M_2 be two models such that $M_1 \subseteq M_2$ and let $e \in M_1 \subseteq M_2$. Assume that Algorithm 13 is applied to M_2 and that e can be removed from M_2 in a recursive call of `FindCheckingModel`. This implies that if Algorithm 13 is applied to M_1, then e can be removed from M_1 in a recursive call of `FindCheckingModel`. This will be the basic property of the algorithm that will be used to prove this theorem.

Let C_b be the checking model of b obtained by Algorithm 13. Let

$$M_b = M_1 \supset M_2 \supset \cdots \supset M_{n-1} \supset M_n = C_b \tag{12.63}$$

be the sequence of sets that is called in the recursive calls. Assume that

$$E_b(\omega_c) \subseteq C_b \tag{12.64}$$

is not true. Then there exists an output C_{b_i} of Algorithm 13 such that $C_{b_i} \setminus C_b \neq \emptyset$ and $C_{b_i} \subseteq M_b$. This implies that there must be a set M_i in (12.63) such that $C_{b_i} \subseteq M_i$ and $C_{b_i} \not\subseteq M_{i+1}$. This means that there is an equation $e \in C_{b_i} \subseteq M_i$, that can be removed from M_i in a recursive call. Since $C_{b_i} \subseteq M_i$, it follows from the basic property of the algorithm, proved above, that e can be removed from C_{b_i} in a recursive call. This implies that C_{b_i} is not an output of Algorithm 13 and this is a contradiction. Hence, (12.64) follows. The inclusions (12.62) and (12.64), imply (12.48). This and Lemma 12.3 imply (12.45) and this completes the proof. □

In conclusion, the isolability predictions proposed in Section 12.2 and Section 12.3 can be computed as $I_s(\omega_{MSO})$ and $I_s(\omega_c)$ respectively.

12.4.1 Isolability Prediction Interpretation

In this section we will exemplify the isolability prediction when using the checking models suggested in both Section 12.2 and in Section 12.3.

Example 12.12
Consider the water-tank example. The analytical isolability of the diagnosis model in Table 12.2 is shown as a diagnosis implication partial order in Figure 12.2. In Section 12.7, we will discuss how the analytical isolability of a diagnosis model can be computed. In Figure 12.2 it can also be seen that no fault is isolable from faults with a superset of faulty components. This is not surprising since no equation in the model holds only in a faulty behavioral mode. Furthermore, since the top element is an upper bound for all behavioral modes, it means that these faults will always be diagnoses, in fact they all have the empty set as their checking models.

The isolability prediction obtained when using $C_b = M_b^+$ as suggested in Section 12.2 is shown as a diagnosis implication partial order in Figure 12.3. As said in Section 12.2, this isolability prediction should be considered as an approximation of the true isolability. For example, **PW** is isolable from **PTQ** according to Figure 12.2, but in the isolability prediction shown in Figure 12.3 **PW** is predicted not to be isolable from **PTQ**. Hence this prediction is not an upper bound for the isolability. However for this example, this prediction is a underestimation of the analytical isolability.

Figure 12.4 shows the diagnosis implication partial order computed by Algorithm 13 in combination with (12.12). This isolability prediction is an upper bound of the isolability. For example the four behavioral modes in the top are an equivalence class and are therefore not isolable from each other. This can be verified in Figure 12.2.

In conclusion, the isolability prediction in Figure 12.4 is more optimistic than the isolability prediction in Figure 12.3.

In the example, we noted that the isolability predictions can be different. The isolability prediction suggested by using PSO sets is the isolability that would be obtained for a linear model in the generic case. The isolability properties contained only in the second isolability prediction are typically isolability properties obtained from non-linearities not contained in the PSO parts of the behavioral models.

12.5 Modeling Guidelines for Good Isolability Predictions

The proposed isolability prediction in Section 12.2 and 12.3 might be different for different diagnosis models describing exactly the same behavior. In this section, two modeling guidelines for obtaining better isolability predictions, will be proposed. To define the first guideline, the following notation will be need.

Recall from Section 2.1 and Section 5.3 that the set of components is denoted by **C**, the set of component behavioral modes for a component $c \in \mathbf{C}$ is denoted by $\mathbf{B}_c$, and the behavioral model for component behavioral mode $b \in \mathbf{B}_c$ is denoted by $M_{c,b}$. As said in Section 2.1.1, a variable in the relations for a

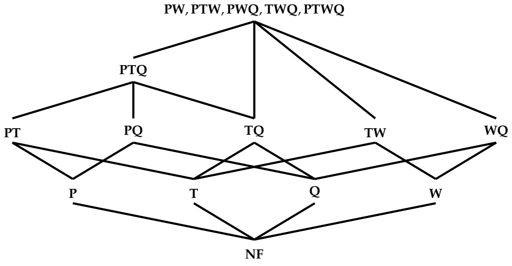

Figure 12.2: The analytical isolability of the diagnosis model in Table 12.2 describing the behavior of the water-tank process.

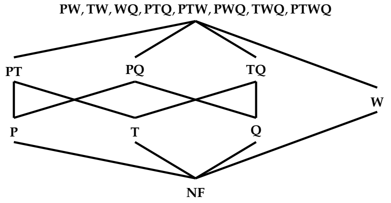

Figure 12.3: An isolability prediction of the water-tank process when PSO sets have been used.

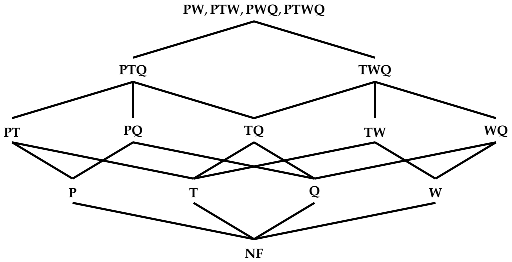

Figure 12.4: An isolability prediction of the water-tank process when Algorithm 13 has been used.

component $c \in \mathbf{C}$ is either an external variable or an internal variable. Let the set of internal variables and the set of external variables be denoted by $\boldsymbol{X}_{ext}$ and $\boldsymbol{X}_{int}$ respectively. The corresponding vectors are denoted by $\boldsymbol{x}_{ext}$ and $\boldsymbol{x}_{int}$ and their corresponding domains by $\mathbb{X}_{ext}$ and $\mathbb{X}_{int}$. Given a component $c \in \mathbf{C}$, let the *external behavior for a component behavioral mode* $b \in \mathbf{B}_c$ be defined as

$$\mathcal{B}_c(b) = \{\boldsymbol{x}_{ext} \in \mathbb{X}_{ext} | \exists \boldsymbol{x}_{int} \in \mathbb{X}_{int} : M_{c,b}(\boldsymbol{x}_{ext}, \boldsymbol{x}_{int})\} \tag{12.65}$$

To define the second guideline the following notation is needed. Consider a model M with the set $\boldsymbol{X}$ of unknown variables and the set $\boldsymbol{Z}$ of known variables. Let $E \subseteq M$ and $X \subseteq \boldsymbol{X} \cup \boldsymbol{Z}$ be arbitrarily chosen such that there exists a perfect matching in the subgraph $G(E, X)$, i.e., $\nu(G(E, X)) = |E| = |X|$. If the model M is a structured linear model, then the matrix corresponding to the equations in E and the variables in X is invertible. In the static case, this means that given values of the variables in $(\boldsymbol{X} \cup \boldsymbol{Z}) \setminus X$, it is possible to compute unique values of the variables in X by using the equations in E. Next we will generalize the notion of structured model to the non-linear static case. A non-linear static model M is said to be *structured* if, for all $E \subseteq M$ and for all $X \subseteq \boldsymbol{X} \cup \boldsymbol{Z}$ such that $\nu(G(E, X)) = |E| = |X|$, there exists for almost all values of the variables in $(\boldsymbol{X} \cup \boldsymbol{Z}) \setminus X$ a unique solution of each variables in X at least locally.

Now, we are ready to state the modeling guidelines:

a) Each component $c \in \mathbf{C}$ and each pair of component behavioral modes $(b_i, b_j) \in \mathbf{B}_c \times \mathbf{B}_c$ should satisfy

$$M_{c,b_j} \subseteq M_{c,b_i} \leftrightarrow \mathcal{B}_c(b_i) \subseteq \mathcal{B}_c(b_j) \tag{12.66}$$

b) For each behavioral mode $b \in \mathbf{B}$, the behavioral model M_b should be structured.

A guideline similar to (b) has been discussed in (Blanke et al., 2003). If it is difficult to make the behavioral models structured, then a less ambitious task is to make the behavioral models for each component behavioral mode structured. The two guidelines (a) and (b) are motivated, discussed, and exemplified next.

12.5.1 Motivation of the Guidelines

Guideline (a) addresses how to select model assumptions, while guideline (b) suggests how to write the model equations on a form suitable for the isolability analysis proposed here.

We start to discuss the motivation of guideline (a). Note that the right implication in (12.66) is trivially true. The important part is therefore to strive for formulating the diagnosis model such that the left implication is fulfilled. In Section 12.1.3 we argued that the checking models should be as small as possible to get the best isolability prediction according to Theorem 12.2. By similar arguments, it follows from (12.11) that the behavioral models M_{b_i} should be as large as possible. By following modeling guideline (a), we can obtain larger behavioral models and this will be illustrated in the next example.

Example 12.13

Consider the model in Table 2.2. This model does not fulfill guideline (a).

Consider for example, the model of the flow sensor component Q:

Assumption	Equation	Expression
$Q = \text{NF}$	e_7	$y_q = q_2$
$Q = \text{B}$	e_8	$y_q = q_2 + f_{yq}$
$Q = \text{B}$	e_9	$\dot{f}_{yq} = 0$

For this component, it holds that

$$\mathcal{B}_Q(\text{NF}) \subseteq \mathcal{B}_Q(\text{B}) \tag{12.67}$$

but

$$M_{Q,\text{B}} = \{e_8, e_9\} \not\subseteq \{e_7\} = M_{Q,\text{NF}} \tag{12.68}$$

However without changing the external behavior of component behavioral mode NF, it is possible to extend the component behavioral model of NF such that

$$M_{Q,\text{NF}} \cup M_{Q,\text{B}} = \{e_7, e_8, e_9\} \tag{12.69}$$

becomes the new component behavioral model $M_{Q,\text{NF}}$. Then, guideline (a) is fulfilled for component Q. This modification is obtained by putting assump(e_8) = assump(e_9) = $\mathbf{B}$. The no-fault behavioral model $M_{\mathbf{NF}}$ is, by this modification, extended with the equations e_8 and e_9. Hence by following guideline (a), behavioral models are extended.

A discussion about how to evaluate subset relations like (12.67) will be postponed to the next section.

Now, we will discuss the purpose of guideline (b). To do this we will consider two cases when guideline (b) is not fulfilled. These two cases correspond to Example 8.7 and Example 8.8 presented in Section 8.3. In Example 8.7, we illustrated for a linear model

$$Hx + Lz = 0 \tag{12.70}$$

that the analytical redundancy can be strictly greater than the structural redundancy if some sub-matrix of H is rank deficient. Furthermore, in Example 8.8, we showed that the analytical redundancy can be strictly less than the structural redundancy, if some sub-matrix of $[H\,L]$ not only contained in H is rank deficient.

We start to exemplify the isolability prediction consequence of the first type of rank deficiency.

Example 12.14

Consider a behavioral model M_b defined as

(12.71)

Equation	Expression
e_1	$y_1 = x_1 + x_2$
e_2	$y_2 = x_1 + x_2$

This model is not structured, since the matrix corresponding to both equations and both unknowns is rank deficient. An example of a non-structured non-linear model M_b is

(12.72)

Equation	Expression
e_1	$y_1 = f(x_1, x_2)$
e_2	$y_2 = f(x_1, x_2)$

where f is an arbitrary non-linear function. If x_1 is eliminated in (12.71) and in (12.72), then we get that $y_1 - y_2 = 0$ in both cases. This means that $\{e_1, e_2\}$ are the minimal checking models for both these behavioral models. For the two behavioral modes (12.71) and (12.72), the PSO parts are the empty set. Hence, the PSO parts are in this case not checking models and the method in Section 12.2 for finding checking models cannot be used.

The next example will illustrate the second type of rank deficiency and its implications on isolability prediction.

Example 12.15
Consider a behavioral model M_b defined as

Equation	Expression
e_1	$y = x$
e_2	$y = x + f$
e_3	$\dot{f} = 0$

(12.73)

This behavioral model does not fulfill modeling guideline (b) and it can be understood as follows. The matrix corresponding to the linear equations $\{e_1, e_2\}$ and the variables $\{y, x\}$ is rank deficient, i.e., this behavioral model is not structured. The set $\{e_1, e_2, e_3\}$ is a PSO set if f and $\dot{f}$ are not distinguished and the minimal checking model for this set is the empty set. For any of the two methods proposed for finding checking models, any PSO set will be contained in the computed checking model. For the example, we see that the checking model for any of the methods will be $\{e_1, e_2, e_3\}$, even if the minimal checking model is the empty set. The computed checking models should be as small as possible according to Section 12.1.3 and for the example, the computed checking models are larger than the minimal checking model. Later we will see that by following guideline (b) smaller checking models can be obtained.

Before the question of how to fulfill the guidelines is addressed, one more example of a behavioral model that do not fulfill guideline (b) is given.

Example 12.16
This example is a continuation of Example 12.13 where equation e_8 and e_9 have been included in the no-fault behavioral model in Table 2.2. The no-fault behavioral model does not fulfill modeling guideline (b) and this can be understood as follows. The polynomial matrix corresponding to the linear equations $\{e_7, e_8\}$ and the variables $\{y_q, q_2\}$ is rank deficient. Hence, the no-fault behavioral mode is not structured.

12.5.2 Fulfilling Guideline (a)

Next we will exemplify how modeling guideline (a) can be fulfilled. Each component $c \in \mathbf{C}$ can be analyzed separately to verify that modeling guideline (a) is fulfilled. To ensure that guideline (a) is fulfilled, relations between the external behavior of different component behavioral modes, of the type shown in (12.67), must be investigated. In Example 12.13 the component models are small, and the relationships between the external behaviors of the component behavioral modes can easily be determined. Often, component behavioral models are small and straightforward computations as described in the example is in these cases manageable. Furthermore, it is common that several components of the same type is used, and then one analysis for one of these components is sufficient.

Assume that we have concluded for a component c and two component behavioral modes b_1 and b_2 that

$$\mathcal{B}_c(b_1) \subseteq \mathcal{B}_c(b_2) \tag{12.74}$$

and

$$M_{c,b_2} \not\subseteq M_{c,b_1} \tag{12.75}$$

Normally, we can fix this by extending M_{c,b_1} with the equations in M_{c,b_2} as in Example 12.13. This is done by modifying assump(e) for some of the equations $e \in \boldsymbol{M}$.

12.5.3 Fulfilling Guideline (b)

Now, we will exemplify how modeling guideline (b) can be fulfilled. We show how this can be done for the water-tank process.

Example 12.17

This is a continuation of Example 12.16. In that example we concluded that the no-fault model in Table 2.2 where e_8 and e_9 have been included does not fulfill modeling guideline (b). The problem was identified to be that the matrix corresponding to the equations in $\{e_7, e_8\}$ and the variables in $\{y_q, q_2\}$ is rank deficient. By introducing a new variable $g = y_q - q_2$ and substitute $y_q - q_2$ for g in e_7 and e_8, we get

Assumption	Equation	Expression
Flow sensor		
$Q = \text{NF}$	e_7	$g = 0$
	e_8	$g = f_{yq}$
	e_9	$\dot{f}_{yq} = 0$
	e_{10}	$g = y_q - q_2$

Since, $g = f_{yq}$ the new variable g can be eliminated and if the equations are reordered and enumerated according to the new order, we obtain

Assumption	Equation	Expression
Flow sensor		
	e_7	$y_q = q_2 + f_{yq}$
$Q = \text{NF}$	e_8	$f_{yq} = 0$
	e_9	$\dot{f}_{yq} = 0$

Now, this part of the no-fault model fulfills modeling guideline (b). Note that this is how a constant bias fault would be modeled by following the FDI-approach described in Section 2.4. If the same ideas are applied to the tank component, the model in Table 12.1 is obtained. By differentiating this model and distinguishing different derivatives, the static model in Table 12.2 is obtained. This model fulfills both guideline (a) and (b).

To conclude this discussion, two modeling guidelines have been proposed and ideas of how to reformulate models to fulfill the guidelines have been presented.

12.6 Applications of Isolability Prediction

Previous sections have described how to compute isolability predictions. Here, we will discuss how isolability predictions can be used for fault modeling specification and for supporting sensor selection to meet given isolability requirements.

12.6.1 Fault Modeling Specification

Assume that safety, maintenance, or legislative requirements state demands on the fault isolability. Given a diagnosis model including fault models, it can be determined by an isolability prediction of the diagnosis model if the proposed fault modeling is insufficient for the fault isolability demands.

Example 12.18
Assume that all double faults must be isolable from each other in the water-tank process. The result shown in Figure 12.4 implies that the isolability demands cannot be fulfilled with the proposed model in Table 12.2. For example no double fault is isolable from **PW**. To make any behavioral mode isolable from **PW**, the behavioral model $M_{\mathbf{PW}}$ must be improved for example by additional fault modeling. The faulty components in **PW** are the pump P and the water-level sensor W and none of these components have fault models.

Assume that it is reasonable to use a constant bias fault model for the water-level sensor. Let f_{yw} be the size of the bias fault. Equation e_5 can now be replaced by $y_w = w + f_{yw}$ and $\dot{e}_5$ by $\dot{y}_w = \dot{w} + \dot{f}_{yw}$ which both hold in any system behavioral-mode. Furthermore, the new equations $e_8 : f_{yw} = 0$ which holds when $W = \text{NF}$, and $\dot{e}_8 : \dot{f}_{yw} = 0$ which always is true, are added to the model in Table 12.2. By applying Algorithm 13 to the model including the new fault model, a smaller set $I_p(\langle C_{b_1}, \ldots, C_{b_n} \rangle)$ is obtained. This means that some faults that were not isolable from some other faults without the fault model, now might become isolable. The result with the additional fault model is that it might become possible to isolate all double faults from all other double faults.

For this example, it is also possible to analyze the true isolability by using the analytical expressions. For example, consider the behavioral modes **PW** and **PT**. Without the additional fault model, **PT** was not isolable from **PW**. When including the fault model, the observation set $\mathcal{O}(M_{\mathbf{PW}})$ for **PW** is defined by

$$\dot{y}_w - 2\, y_q\, \dot{y}_q = 0 \tag{12.76}$$

and $\mathcal{O}(M_{\mathbf{PT}})$ is defined by

$$\begin{aligned} \dot{y}_w\, y_q - 2\, y_w\, \dot{y}_q &= 0 \\ y_q = 0 \rightarrow y_w &= \dot{y}_w = 0 \end{aligned}$$

Both these expressions can be computed by elimination of all unknowns in their corresponding checking models respectively. The mode **PT** is isolable from **PW** if $\mathcal{O}(M_{\mathbf{PT}}) \setminus \mathcal{O}(M_{\mathbf{PW}}) \neq \emptyset$. An example of observations in $\mathcal{O}(M_{\mathbf{PT}}) \setminus \mathcal{O}(M_{\mathbf{PW}})$ is $y_q \neq 0, \dot{y}_q \neq 0, y_w \neq y_q^2$, and $\dot{y}_w = 2y_w\, \dot{y}_q / y_q$. Hence $\mathcal{O}(M_{\mathbf{PT}}) \setminus \mathcal{O}(M_{\mathbf{PW}}) \neq \emptyset$, i.e., **PT** is isolable from **PW**. According to the result of Algorithm 13 and the isolability prediction $I_p(\langle C_{b_1}, \ldots, C_{b_n} \rangle)$, it is possible that all double faults are isolable from all other double faults and by pairwise comparisons of observation sets as in the discussion above, it can be shown to be so.

12.6.2 Design Alternative Evaluation

Suppose there are different design alternatives, e.g. different possible sensor configurations. Since only a structural model is needed as input to predict the isolability using any of the two methods presented Section 12.2 and in Section 12.3, the isolability aspects of different design alternatives can easily be evaluated.

Example 12.19

Consider again the water-tank process. Let the isolability demands be the same as in the previous section and assume that there are two design alternatives for the water-tank process, one as previously described in Section 12.5 and one with an additional flow sensor Q_{extra} measuring q_1. We know from the previous discussion that it is not possible to isolate all double faults from each other by using the model in Table 12.2. The result of applying Algorithm 13 to an extended model including the additional sensor Q_{extra} and computing an isolability prediction, addresses the question if the model with the additional sensor can be sufficient to meet the isolability demands.

The extended model is obtained by adding the equation $e_8 : y = q_1$ with the assumption $Q_{\text{extra}} = \text{NF}$. Note that an extra sensor will change the set of all system behavioral modes. In this example, the number of components is 5 and the original model has only 4 components. By including the additional sensor, all double faults, including the new once introduced by Q_{extra}, might be isolable from any other double fault according to the isolability prediction. Analytical analysis can be done as in Section 12.6.1 to conclude that all double faults are isolable from all other double faults.

To summarize the results of Example 12.18 and 12.19, without any fault model or any additional sensor, this analysis shows that there are double faults which are not isolable from other double faults. However, by adding the proposed fault model or the water-level sensor it can be shown that all double faults are isolable from all other double faults.

12.7 Designing a Diagnosis System with Maximum Isolability

In this section, we will combine isolability predictions with the test selection method presented in Section 11.9 to develop a design procedure for deriving a diagnosis system with the maximum possible isolability given a diagnosis model. The basic idea is to consider a large set of potential test corresponding to different sub-models, primarily MSO sets, of the diagnosis model. The set of models will be selected such that if tests are constructed for all sets, the resulting diagnosis system has maximum isolability. To ensure this, we will in addition to the MSO sets include checking models for all behavioral modes in the considered set of models. To reduce test construction time, that often involves manual work, and the on-line computational burden using the diagnosis system, the number of tests will be minimized. This will be done by selecting a minimum cardinality subset of the potential test with the maximum possible isolability. Then only the selected tests are constructed. Before describing the design procedure, we will first specify what we mean by maximum possible isolability given a diagnosis model.

12.7.1 Characterizing Maximum Isolability

One might think that for any complete diagnosis system Δ based on a diagnosis model $\mathbb{M}$, it holds that the isolability of the diagnosis model is an upper bound for the isolability of the diagnosis system, i.e., $I(\Delta) \subseteq I(\mathbb{M})$. However, the following example shows that this is not true in general.

Example 12.20
Consider a diagnosis model $\mathbb{M}$ specified by the following observation sets:

$$\begin{aligned} \mathcal{O}(M_{b_1}) &= \{0\} \\ \mathcal{O}(M_{b_2}) &= \{0, 1\} \\ \mathcal{O}(M_{b_3}) &= \{0, 1, 2\} \end{aligned} \tag{12.77}$$

It is clear that the analytical isolability of this model $\mathbb{M}$ is

$$I(\mathbb{M}) = \{(b_3, b_2), (b_3, b_1), (b_2, b_1)\} \tag{12.78}$$

Consider a complete diagnosis system $\Delta = \{\delta\}$, where δ is defined by the acceptance set $\mathcal{O}(\delta) = \{0, 1\}$ and $\Phi = \{b_2\}$. Note that $\Phi^C = \{b_1, b_3\}$ is, in this case, not equal to the set $\{b_3\}$ of faults that influence the test. If $z = 2$, then the null hypothesis of the test is rejected and the conclusion drawn is that $\text{sys} \in \Phi^C = \{b_1, b_3\}$ and not in b_2. This means that the analytical isolability of the diagnosis system is

$$I(\Delta) = \{(b_3, b_2), (b_1, b_2)\} \tag{12.79}$$

By comparing (12.78) with (12.79), it can be seen that $(b_1, b_2) \in I(\Delta)$ and $(b_1, b_2) \notin I(\mathbb{M})$. Hence by analyzing the candidates computed by the diagnosis system we falsely believe that b_1 is isolable from b_2.

In the example we saw that $I(\Delta) \subseteq I(\mathbb{M})$ is not true for a complete diagnosis system Δ in general. This was due to the fact that the candidates generated false isolability properties. When maximizing the analytical isolability of a diagnosis system based on a model $\mathbb{M}$, we do not want to maximize the number of false isolability properties. In the design procedure, that will be describe later in Section 12.7.4, we will consider diagnosis tests where the set Φ_i^C is equal to the set of behavioral modes that influences the tests. In this way, we will according to the next theorem avoid the problem with false isolability.

Theorem 12.9. *Given a diagnosis system Δ, where for each $\delta_i \in \Delta$ the set Φ_i^C is equal to the set of behavioral modes that influences δ_i, it holds that*

$$I(\Delta) \subseteq I(\mathbb{M}) \tag{12.80}$$

Proof. Take an arbitrary $(b_1, b_2) \in I(\Delta)$. Theorem 11.1 implies that $(b_1, b_2) \in I_s(\Delta)$. This means according to (11.11) that there exists a test $\delta_i \in \Delta$ such that

$$b_1 \notin \Phi_i \tag{12.81}$$

and

$$b_2 \in \Phi_i \tag{12.82}$$

Since the set Φ_i^C is equal to the set of behavioral modes that influences δ_i, it follows from (12.81), that

$$\mathcal{O}(M_{b_1}) \not\subseteq \mathcal{O}(\delta_i) \tag{12.83}$$

and from (12.82), that

$$\mathcal{O}(M_{b_2}) \subseteq \mathcal{O}(\delta_i) \tag{12.84}$$

From (12.83) and (12.84), we get that

$$\mathcal{O}(M_{b_1}) \setminus \mathcal{O}(M_{b_2}) \neq \emptyset \tag{12.85}$$

This means that there exists an observation $z_0 \in \mathcal{Z}$ such that

$$z_0 \in \mathcal{O}(M_{b_1}) \wedge z_0 \notin \mathcal{O}(M_{b_2}) \tag{12.86}$$

This is according to (2.5) equivalent to

$$b_1 \in \mathcal{D}(z_0) \wedge b_2 \notin \mathcal{D}(z_0) \tag{12.87}$$

This and the definition of analytical isolability of a diagnosis model in (12.1) imply that $(b_1, b_2) \in I(\mathbb{M})$. Since (b_1, b_2) was arbitrarily chosen in $I(\Delta)$, the theorem follows. □

12.7.2 Lower and Upper Bounds for the Maximum Isolability

By assuming that the set Φ_i^C is equal to the set of behavioral modes that influences test δ_i, the maximum isolability of any diagnosis system based on a diagnosis model $\mathbb{M}$ is according to Theorem 12.9 equal to $I(\mathbb{M})$. The task of deriving a diagnosis system with maximum isolability can therefore be formulated as the intuitive task to find a diagnosis system with isolability equal to $I(\mathbb{M})$. As said before, $I(\mathbb{M})$ is in general unknown and analytically difficult to derive.

However, it is possible to compute a lower and an upper bound of $I(\mathbb{M})$. An upper bound will be obtained by using a set of potential tests Δ_a such that

$$I(\mathbb{M}) \subseteq I_s(\Delta_a) \tag{12.88}$$

How to select Δ_a such that (12.88) holds is discussed later in Section 12.7.3. A set $\Delta \subseteq \Delta_a$ is selected such that

$$I_s(\Delta_a) = I_s(\Delta) \tag{12.89}$$

If the fault influence for each $\delta_i \in \Delta$ is equal to the set Φ_i^C, it follows according to Theorem 11.3 that

$$I_s(\Delta) = I(\Delta) \tag{12.90}$$

and according to Theorem 12.9 that

$$I(\Delta) \subseteq I(\mathbb{M}) \tag{12.91}$$

The expressions (12.88)-(12.90) imply that

$$I(\mathbb{M}) = I_s(\Delta_a) = I_s(\Delta) = I(\Delta) \tag{12.92}$$

that is, Δ is a diagnosis system with the maximum isolability $I(\mathbb{M})$.

12.7.3 Selecting the Set of Potential Tests Δ_a

The set Δ_a of potential tests must be selected carefully to fulfill (12.88) and how to do this will be described in this section. Let $\omega_c = \{C_{b_1}, \ldots, C_{b_n}\}$ be the set of one checking models for each behavioral mode $b_i \in \mathbf{B}$. This set can be computed by using Algorithm 13. Let $\Delta_c = \{\delta_1, \ldots, \delta_n\}$ be a diagnosis system such that $\mathcal{O}(\delta_i) = \mathcal{O}(C_{b_i}) = \mathcal{O}(M_{b_i})$ and let Φ_i be any sets such that Δ_c is complete. Then Δ_c is also sound according to Theorem 2.2. Since Δ_c is a sound and complete diagnosis system the following theorem implies that Δ_c has the maximum possible isolability.

Theorem 12.10. *If Δ is a sound and complete diagnosis system with respect to a diagnosis model $\mathbb{M}$, it follows that*

$$I(\mathbb{M}) = I(\Delta) \tag{12.93}$$

Proof. Since Δ is a sound and complete diagnosis system with respect to a diagnosis model $\mathbb{M}$, it means that

$$\forall z \in \mathbb{Z} : (\mathcal{D}(z) = \mathcal{C}(z)) \tag{12.94}$$

Definition 12.3 implies that

$$I(\mathbb{M}) = \{(b_1, b_2) | \exists z \in \mathbb{Z} : \left(b_1 \in \mathcal{D}(z) \wedge b_2 \notin \mathcal{D}(z)\right)\} \tag{12.95}$$

From (12.94) and (12.95), it follows that

$$I(\mathbb{M}) = \{(b_1, b_2) | \exists z : \left(b_1 \in \mathcal{C}(z) \wedge b_2 \notin \mathcal{C}(z)\right)\} \tag{12.96}$$

Expression (11.1) in Definition 11.3 and (12.96) imply (12.93). □

Since Δ_c is a sound and complete diagnosis system, Theorem 12.10 implies that

$$I(\mathbb{M}) = I(\Delta_c) \tag{12.97}$$

From this and Theorem 11.1, we get that

$$I(\mathbb{M}) \subseteq I_s(\Delta_c) \tag{12.98}$$

If the set Δ_a in (12.88) is chosen as $\Delta_c \subseteq \Delta_a$, then (12.88) is implied by (12.98). The reason for not choosing $\Delta_c = \Delta_a$ is that our primarily choice will be to construct tests based on MSO sets. The checking models are only included in Δ_a to ensure that if all tests in Δ_a are constructed a diagnosis system with maximum isolability is obtained. Note that, even if observation sets $\mathcal{O}(C_{b_i})$ are mentioned in the derivation of (12.88), we do not need to compute any observation set explicitly to know that (12.88) is true. In fact, it is sufficient to know that ω_c contains a checking model for each behavioral mode.

Now, we have described how to choose the set of potential tests Δ_a. Next, we will describe how to select a minimum cardinality set of tests $\Delta \subseteq \Delta_a$ with the maximum possible isolability.

12.7.4 Design Procedure

The test selection will be done by the following design procedure for constructing a diagnosis system with the maximum isolability given a diagnosis model $\mathbb{M}$.

1. Find all feasible MSO sets ω_{MSO} by using Algorithm 7 and compute checking models ω_c by using for example Algorithm 13. Let $\omega_a = \omega_{MSO} \cup \omega_c$.

2. Let Δ_a denote a set of potential tests, one for each model in ω_a. Let the test assumption of $\delta_i \in \Delta_a$ be given by $\Phi_i = \text{assump}(M_i)$, where M_i is the corresponding model in ω_a.

3. Find a diagnosis system based on a minimal set $\Delta \subseteq \Delta_a$ such that $I_s(\Delta) = I_s(\Delta_a)$. This step has been described in Section 11.9.4.

4. Construct the ideal tests $\delta_i \in \Delta$ such that $\mathcal{O}(\delta_i) = \mathcal{O}(M_i)$.

5. Evaluate the set of faults that influence each test in Δ.

6. If the fault influence is not equal to Φ_i^C for some test in Δ, then modify the sets Φ_i for the tests in Δ_a and goto step (3).

7. If the fault influence is equal to Φ_i^C for all test in Δ, then Δ is a diagnosis system with the maximum possible isolability, i.e., $I(\Delta) = I(\mathbb{M})$.

Note that the steps (3)-(7) corresponds to the design procedure presented in Section 11.9.4 for finding a diagnosis system with the maximum isolability given a set of potential tests. To ensure that the obtained diagnosis system has maximum possible isolability, $I(\Delta) = I(\mathbb{M})$, we include $\omega_c \subseteq \omega_a$ in step (1) and the acceptance set is related to the diagnosis model by $\mathcal{O}(\delta_i) = \mathcal{O}(M_i)$ according to step (4). Computing observations sets is a difficult problem. However the requirement that $\mathcal{O}(\delta_i) = \mathcal{O}(M_i)$ in step (4) can be omitted for all models $M_i \notin \omega_c$, because ideal tests only for all checking models have the maximum possible isolability. Since the difficult requirement can be omitted in step (4) if the select tests are not based on checking models, we will prefer to select tests based on other models. Note also that ω_a can be selected as any superset of ω_c.

We will illustrate the design procedure on the water-tank example.

Example 12.21
Consider the diagnosis model in Table 12.2 and assume that we want to derive a diagnosis system with the maximum possible isolability by following the design procedure. In step (1), all 11 feasible MSO sets ω_{MSO} contained in this model are found by using Algorithm 11. Furthermore, the set of all checking models ω_c are computed by using Algorithm 13. In this case, there are 12 different non-empty checking models of all 16 behavioral modes.

In step (2), a set of potential test Δ_a is defined, one for each model in $\omega_{MSO} \cup \omega_c$.

In step (3), the structural isolability $I_s(\Delta_a) = I_p(\langle C_{b_1}, \ldots, C_{b_n} \rangle)$ is computed. This structural isolability is equal to the one shown in Figure 12.4. There are 4 test sets $\Delta \subseteq \Delta_a$ such that $I_s(\Delta) = I_s(\Delta_a)$ and with the minimum cardinality 6. We choose to maximize the number of tests based on MSO sets and of the four minimum cardinality test sets there is only one which includes 2 MSO

Table 12.4: Tests based on MSO set in the model shown in Table 12.2.

	b	Set	Test Assumption
1		$\{e_3, \dot{e}_3, \dot{e}_4, e_5, \dot{e}_5, e_6, \dot{e}_6, e_7, \dot{e}_7\}$	$W = NF \land Q = NF$
2		$\{e_3, \dot{e}_3, e_4, \dot{e}_4, e_5, \dot{e}_5, \dot{e}_6, \dot{e}_7\}$	$T = NF \land W = NF$
3	**TW**	$\{e_1, e_2, \dot{e}_3, \dot{e}_4, e_6, \dot{e}_6, e_7, \dot{e}_7\}$	$P = NF \land Q = NF$
4	**WQ**	$\{e_1, e_2, \dot{e}_3, e_4, \dot{e}_4, \dot{e}_6, \dot{e}_7\}$	$P = NF \land T = NF$
5	**PTQ**	$\{e_3, \dot{e}_3, \dot{e}_4, e_5, \dot{e}_5, \dot{e}_6, \dot{e}_7\}$	$W = NF$
6	**TWQ**	$\{e_1, e_2, \dot{e}_3, \dot{e}_4, \dot{e}_6, \dot{e}_7\}$	$P = NF$
7		$\{e_1, e_2, e_3, \dot{e}_3, \dot{e}_4, e_5, \dot{e}_5, \dot{e}_6, \dot{e}_7\}$	$P = NF \land W = NF$

sets. The six chosen tests correspondence to the six first rows in Table 12.4. The first two correspond to the MSO sets while the next four rows correspond to the checking models. The acceptance sets for the checking models can be computed by using quantifier elimination tools in Mathematica.

In step (4), the acceptance set is computed for the selected tests in Δ. These are shown in Table 12.5. Note, that the acceptance set for the checking model of **TWQ** is equal to $\mathbb{Z}$. This means that the null hypothesis cannot be rejected and that the empty set of behavioral modes influence this test.

In step (5), the fault influence is investigated. The set of fault that influence each test is equal to Φ_i^C for all test in Δ, except for test δ_6 based on the checking model of **TWQ**.

In step (6), the test assumption Φ_6 for test δ_6 is modified and set to $\Phi_6 = \mathbf{B}$. Then, we go back to step (3) in the design procedure.

By using the modified test assumption, the structural isolability $I_s(\Delta_a)$ is shown in Figure 12.2. There are 6 minimum cardinality test sets with the maximum structural isolability. Of these test sets, there is only one set that contains 3 tests based on MSO sets. The other 5 test sets contains fewer test based on MSO sets. Let the set with maximum number of tests based on MSO sets be the set of selected tests Δ. This set corresponds to the rows 1, 2, 3, 4, 5, and 7 in Table 12.4 and in Table 12.5. Note that compared to the previously selected tests only one additional test δ_7 is selected, i.e., the test that corresponds to row 7.

The acceptance set and the set of faults that influence test δ_7 is computed in step (4) and step (5) respectively. The set of faults that influence each test is equal to Φ_i^C and this means according to step (7) that a diagnosis system with the maximum possible isolability $I(\Delta) = I(\mathbb{M})$ has be found. Note that, in addition to finding a diagnosis system with the maximum possible isolability, we have also computed the isolability in the diagnosis model $\mathbb{M}$, which until now has been unknown.

12.8 Conclusions

In the development of processes including diagnosis, design decisions are taken, e.g. sensor configuration selection, which affects the fault isolability possibilities. This chapter has presented a method to compute isolability pre-

Table 12.5: Tests based on MSO set in the model shown in Table 12.2.

	$z \in \mathcal{O}(\delta)$
1	$\dot{y}_w = 2\dot{y}_q y_w / y_q$
2	$\dot{y}_w^2 = 4\dot{y}_q^2 y_w$
3	$(\dot{y}_q = 0 \wedge u = y_q) \vee (\dot{y}_q \neq 0 \wedge (u = 0 \vee y_q \neq 0))$
4	$\dot{y}_q = -1/2 \rightarrow u = 0$
5	$\dot{y}_w = 0 \leftrightarrow (y_q = 0 \vee y_w = 0)$
6	$z \in \mathbb{Z}$
7	$\dot{y}_w^2 = u\dot{y}_w - 2y_w\dot{y}_q$

dictions that easily can computed for different design alternatives and evaluate their isolability limitations. It has also been shown how the proposed isolability predictions can be used when designing a diagnosis system with maximum possible isolability.

Here two different isolability predictions have been proposed. The prediction obtained by following the description in Section 12.2, that is to use PSO parts as checking models, gives the same result as in (Travé-Massuyès et al., 2003; Frisk et al., 2003), and (Blanke et al., 2003). Under the assumption that all monitorable equations are contained in the PSO part, this prediction is an upper bound for the true isolability.

The second type of isolability prediction presented in Section 12.3 is more conservative. The advantage of being conservative is that the second isolability prediction is always an upper bound for the true isolability in contrast to the results in (Travé-Massuyès et al., 2003), (Frisk et al., 2003), and (Blanke et al., 2003) where analytical assumptions need to be satisfied. This prediction was done by combining Algorithm 13, which computes the smallest checking models that can be computed by using structural models as the one in Table 12.3, and the link between checking models and isolability stated in Theorem 12.2.

Different applications for isolability predictions have been give in Section 12.6 and in Section 12.7. In Section 12.6.1, it was shown how a fault prediction could detect insufficient fault modeling. The analysis revealed faults not isolable from other faults, and by the example a methodology was proposed to locate required additional fault modeling. Section 12.6.2 showed how a fault prediction could be used to find the isolability limitations of different design alternative for a process to be diagnosed. Finally, Section 12.7 presented a design procedure for finding a diagnosis system with the maximum possible isolability. This design procedure was based on isolability predictions in combination with the theory for selecting tests presented in Chapter 11. In conclusion, it is believed that structural methods for isolability analysis have an advantage of analytical methods to support decisions early in the design process and also for test selection during diagnosis system construction.

R

H. Alt, N. Blum, K. Mehlhorn, and M. Paul. Computing a maximum cardinality matching in a bipartite graph in time $O(n^{1.5}\sqrt{m/\log n})$. *Information Processing Letters*, 37(4):237–240, 1991.

J. Åslund, J. Biteus, E. Frisk, M. Krysander, and L. Nielsen. A systematic inclusion of diagnosis performance in fault tree analysis. In *IFAC World Congress*, Prague, Czech Republic, 2005.

J. Åslund, J. Biteus, E. Frisk, M. Krysander, and L. Nielsen. Safety analysis of autonomous systems by extended fault tree analysis. *International Journal of Adaptive Control and Signal Processing*, 2006. Accepted for publication.

A. Asratian, T. Denley, and R. Häggkvist. *Bipartite Graphs and their Applications*. Cambridge University Press, 1998.

M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes - Theory and Application*. Prentice Hall, 1993.

M. Blanke, M. Kinneart, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer-Verlag, 2003.

B. Buchberger. *An Algorithm for Finding a Basis for the Residue Class Ring of a Zero-dimensional Polynomial Ideal*. PhD thesis, University of Innsbruck, Austria, 1965.

G. Casella and R. L.Berger. *Statistical Inference*. Duxbury Press, 1990.

J. P. Cassar and M. Staroswiecki. A structural approach for the design of failure detection and identification systems. In *IFAC Control of Industrial Systems*, Belford, France, 1997.

J. Chen and R. J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer, 1999.

E. Y. Chow and A.S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Trans. on Automatic Control*, 29(7):603–614, 1984.

M.-O. Cordier, P. Dague, F. Levy J. Montmain, M. Staroswiecki, and L. Trave-Massuyes. Possible conflicts: a compilation technique for consistency-based diagnosis approach from the artificial intelligence and automatic control perspectives. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, 34(5):2163– 2177, 2004.

D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer–Verlag, second edition, 1997.

O. Dressler and P. Struss. Model-based diagnosis with the default-based diagnosis engine: Effective control strategies that work in practice. In *In Proceedings of the European Conference on Artificial Intelligence*, pages 677–681. ECAI-94, John Wiley & Sons, 1994.

O. Dressler and P. Struss. A toolbox integrating model-based diagnosability analysis and automated generation of diagnostics. In *Proceedings of the 14th International Workshop on Principles of Diagnosis (DX03)*, pages 99–104, Washington, USA, 2003.

O. Dressler, C. Böttcher, M. Montag, and A. Brinkop. Qualitative and quantitative models in a model-based diagnosis system for ballast tank systems. In *Int. Conf. on Fault Diagnosis (TOOLDIAG)*, pages 397–405, Toulouse, France, 1993.

A. L. Dulmage and N. S. Mendelsohn. Coverings of bipartite graphs. *Canadian Journal of Mathematics*, 10:517–534, 1958.

D. Dustegör, E. Frisk, V. Coquempot, M. Krysander, and M. Staroswiecki. Structural analysis of fault isolability in the DAMADICS benchmark. *Control Engineering Practice*, 14(6):597–608, 2006.

L. Eriksson. Structural algorithms for diagnostic system design using simulink models. Master's thesis, Linköpings universitet, 2004.

P. M. Frank. On-line fault detection in uncertain nonlinear systems using diagnostic observers: a survey. *International Journal of Systems Science*, 25(12): 2129–2154, 1994.

E. Frisk. *Residual Generation for Fault Diagnosis*. PhD thesis, Linköpings Universitet, November 2001.

E. Frisk and M. Nyberg. Using minimal polynomial bases for fault diagnosis. In *Proc. European Control Conference*, 1999.

E. Frisk, D. Düştegör, M. Krysander, and V. Cocquempot. Improving fault isolability properties by structural analysis of faulty behavior models: application to the DAMADICS benchmark problem. In *Proceedings of IFAC Safeprocess'03*, Washington, USA, 2003.

E. Frisk, M. Krysander, M. Nyberg, and J. Åslund. A toolbox for design of diagnosis systems. In *Proceedings of IFAC Safeprocess'03*, Beijing, P. R. China, 2006.

E. Frisk and J. Åslund. Lowering orders of derivatives in non-linear residual generation using realization theory. *Automatica*, 41(10):1799–1807, 2005.

J. Gertler. Analytical redundancy methods in fault detection and isolation; survey and synthesis. In *IFAC Fault Detection, Supervision and Safety for Technical Processes*, pages 9–21, Baden-Baden, Germany, 1991.

J. Gertler and D. Singer. A new structural framework for parity equation-based failure detection and isolation. *Automatica*, 26(2):381–388, 1990.

J. J. Gertler. *Fault Detection and Diagnosis in Enginerering Systems*. Marcel Dekker, Inc., 1998.

J. L. Gross and J. Yellen. *Handbook of Graph Theory*. CRC Press, 2003.

W. Hamscher, L. Console, and J. de Kleer, editors. *Readings in Model-Based Diagnosis*. Morgan Kaufmann Publishers, 1992. ISBN 1-55860-249-6.

F. Harary. *Graph theory*. Addison-Wesley publishing company, ISBN 0-201-41033-8, 1969.

R. Isermann. Fault diagnosis of machines via parameter estimation and knowledge processing - tutorial paper. *Automatica*, 29(4):815–835, 1993.

R. Isermann and P. Balle. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5 (5):709–719, 1997.

R. Izadi-Zamanabadi and M. Blanke. Structural analysis for diagnosis with application to ship propulsion problem. In *IFAC World Congress*, July 2002.

M. Jirstrand. Nonlinear control design by quantifier elimination. *Journal of Symbolic Computation*, 24(2):137–152, August 1997.

M. Jirstrand. *Constructive Methods for Inequality Constraints in Control*. PhD thesis, Linköping University, 1998.

T. Kailath. *Linear Systems*. Prentice-Hall, 1980.

R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. New York: Plenum Press, 1972.

M. Kinneart. Robust fault detection based on observers for bilinear systems. *Automatica*, 35(11):1829–1842, 1999.

J. De Kleer and B.C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.

J. De Kleer, A. K. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56, 1992.

J. Korbicz, J.M. Koscielny, Z. Kowalczuk, and W. Cholewa. *Fault Diagnosis*. Springer, 2004.

M. Krysander. Design and analysis of diagnostic systems utilizing structural methods. Technical Report ISBN 91-7373-733-X, ISSN 0280-7971, Electrical Engineering, 2003. LiU-TEK-LIC-2003:37, Thesis No. 1038.

M. Krysander and J. Åslund. Graph theoretical methods for finding analytical redundancy relations in overdetermined differential algebraic systems. In *IMACS World Congress*, Paris, France, 2005.

M. Krysander and M. Nyberg. Structural analysis utilizing MSS sets with application to a paper plant. In *Proc. of the Thirteenth International Workshop on Principles of Diagnosis*, Semmering, Austria, May 2002a.

M. Krysander and M. Nyberg. Structural analysis for fault diagnosis of DAE systems utilizing MSS sets. In *IFAC World Congress*, Barcelona, Spain, July 2002b.

M. Krysander and M. Nyberg. Fault isolability prediction of diagnostic models. In *16th International Workshop on Principles of Diagnosis (DX-05)*, Pacific Grove, California, USA, 2005.

M. Krysander and M. Nyberg. Fault diagnosis utilizing structural analysis. In *CCSSE*, October 2002c.

M. Krysander, J. Åslund, and M. Nyberg. An efficient algorithm for finding over-constrained sub-systems for construction of diagnostic tests. In *16th International Workshop on Principles of Diagnosis (DX-05)*, Pacific Grove, California, USA, 2005.

T. Lorentzen, M. Blanke, and H. Niemann. Structural analysis - a case study of the rømer satellite. In *Proceedings of IFAC Safeprocess'03*, Washington, USA, 2003.

E. Mansfield. *Differential Gröbner Bases*. PhD thesis, University of Sydney, 1991.

M. A. Massoumnia, G. C. Verghese, and A. S. Willsky. Failure detection and identification. *IEEE Trans. on Automatic Control*, 34(3):316–321, 1989.

R. Mattone and A. De Luca. Relaxed fault detection and isolation: An application to a nonlinear case study. *Automatica*, 42(1):109–116, 2006.

B. Mishra. *Algorithmic Algebra (Texts & Monographs in Computer Science)*. Springer, 1993.

K. Murota. *Matrices and Matroids for System Analysis*. Springer-Verlag, 2000.

R. Nikoukhah. A new methodology for observer design and implementation. *IEEE Trans. on Automatic Control*, 43(2):229–234, 1998.

M. Nyberg. Automatic design of diagnosis systems with application to an automotive engine. *Control Engineering Practice*, 7(8):993–1005, 1999.

M. Nyberg. Model-based diagnosis of an automotive engine using several types of fault models. *IEEE Transaction on Control Systems Technology*, 10(5): 679–689, September 2002a.

M. Nyberg. A fault isolation algorithm for the case of multiple faults and multiple fault types. In *Proceedings of IFAC Safeprocess'03*, Beijing, P. R. China, 2006.

M. Nyberg. Criterions for detectability and strong detectability of faults in linear systems. *International Journal of Control*, 75(7):490–501, May 2002b.

M. Nyberg and E. Frisk. Residual generation for fault diagnosis of systems described by linear differential-algebraic equations. *IEEE Transactions on Automatic Control*, 2006. Accepted for publication.

M. Nyberg and M. Krysander. Combining AI, FDI, and statistical hypothesis-testing in a framework for diagnosis. In *Proceedings of IFAC Safeprocess'03*, Washington, USA, 2003.

O. Ore. Studies on directed graphs i. *Annals of Mathematics*, 63(383–406), 1956.

R. Patton, P. Frank, and R. Clark, editors. *Fault diagnosis in Dynamic systems*. Systems and Control Engineering. Prentice Hall, 1989.

R. J. Patton, P. M. Frank, and R.N. Clark. *Issues of Fault Diagnosis for Dynamic Systems*. Springer, 2000.

C. De Persis and A. Isidori. A geometric approach to nonlinear fault detection and isolation. *IEEE Transactions on Automatic Control*, 46(6):853–865, 2001.

S. Ploix, M. Desinde, and S. Touaf. Automatic design of detection tests in complex dynamic systems. In *Proceedings of 16th IFAC World Congress, Prague*, Prague, Czech Republic, 2005.

J. W. Polderman and J. C. Willems. *Introduction to Mathematical Systems Theory: A Behavioral Approach*. Springer-Verlag, 1998.

B. Pulido and C. Alonso. Possible conflicts, ARRs, and conflicts. In M. Stumptner and F. Wotawa, editors, *Proceedings DX-2002*, pages 122–127, Semmering, Austria, 2002.

B. Pulido and C. A. Gonzalez. Possible conflicts: a compilation technique for consistency-based diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, 34(5):2192– 2206, 2004.

L. Rattfält. A comparative study of two structural methods for fault isolability analysis. Master's thesis, Linköpings Universitet, SE-581 83 Linköping, 2004. URL:http://www.vehicular.isy.liu.se/Publications/.

R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32: 57–95, 1987.

J. Ritt. *Differential Algebra*. Dover Publications, 1950.

L. Råde and B. Westergren. *Beta Mathematics Handbook*. Studentlitteratur, Lund, third edition edition, 1992.

S. Skiena. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. Reading, MA: Addison-Wesley, 1990.

M. Stamatelatos and W. Vesley. Fault tree handbook with aerospace applications. Technical report, NASA, U.S.A., 2002.

D. H. Stamatis. *Failure Mode and Effect Analysis: FMEA from Theory to Execution*. ASQ Quality Press, Milwaukee, Wisconsin, 1995. ISBN 0-87389-300-X.

M. Staroswiecki and G. Comtet-Varga. Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, 37(5): 687–699, 2001.

L. Travé-Massuyès, T. Escobet, and R. Milne. Model-based diagnosability and sensor placement. application to a frame 6 gas turbine subsystem. In D. T. Dupre' S. Mcilraith, editor, *DX01 twelfth international workshop on principles of diagnosis*, pages 205–212, 2001.

L. Travé-Massuyès, T. Escobet, and S. Spanache. Diagnosability analysis based on component supported analytical redundancy relations. In *Proceedings of IFAC Safeprocess'03*, Washington, USA, 2003.

W. Vesley, Goldberg, Roberts, and Haasl. Fault tree handbook. Technical Report NUREG-0492, U.S. N.R.C., U.S.A., 1981.

A. D. Wittkopf. *Algorithms and Implementations for Differential Elimination*. PhD thesis, Simon Fraser University, Burnaby, British Columbia, Canada, 2004.

A. D. Wittkopf and G. J. Reid. Fast differential elimination in c: The cdiffelim environment. *Computer Physics Communications*, 139(2):192–217, 2001.

D. Yu and D.N. Shields. A bilinear fault detection filter. *International Journal of Control*, 68(3):417–430, 1997.