Advances in Industrial Control

Chunlei Zhang
Raúl Ordóñez

Extremum-Seeking Control and Applications

A Numerical Optimization-Based Approach

AIC

Springer

Advances in Industrial Control

For further volumes:
www.springer.com/series/1412

Other titles published in this series:

Digital Controller Implementation and Fragility
Robert S.H. Istepanian and James F. Whidborne (Eds.)

Optimisation of Industrial Processes at Supervisory Level
Doris Sáez, Aldo Cipriano and Andrzej W. Ordys

Robust Control of Diesel Ship Propulsion
Nikolaos Xiros

Hydraulic Servo-systems
Mohieddine Mali and Andreas Kroll

Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques
Silvio Simani, Cesare Fantuzzi and Ron J. Patton

Strategies for Feedback Linearisation
Freddy Garces, Victor M. Becerra, Chandrasekhar Kambhampati and Kevin Warwick

Robust Autonomous Guidance
Alberto Isidori, Lorenzo Marconi and Andrea Serrani

Dynamic Modelling of Gas Turbines
Gennady G. Kulikov and Haydn A. Thompson (Eds.)

Control of Fuel Cell Power Systems
Jay T. Pukrushpan, Anna G. Stefanopoulou and Huei Peng

Fuzzy Logic, Identification and Predictive Control
Jairo Espinosa, Joos Vandewalle and Vincent Wertz

Optimal Real-time Control of Sewer Networks
Magdalene Marinaki and Markos Papageorgiou

Process Modelling for Control
Benoît Codrons

Computational Intelligence in Time Series Forecasting
Ajoy K. Palit and Dobrivoje Popovic

Modelling and Control of Mini-Flying Machines
Pedro Castillo, Rogelio Lozano and Alejandro Dzul

Ship Motion Control
Tristan Perez

Hard Disk Drive Servo Systems (2nd Ed.)
Ben M. Chen, Tong H. Lee, Kemao Peng and Venkatakrishnan Venkataramanan

Measurement, Control, and Communication Using IEEE 1588
John C. Eidson

Piezoelectric Transducers for Vibration Control and Damping
S.O. Reza Moheimani and Andrew J. Fleming

Manufacturing Systems Control Design
Stjepan Bogdan, Frank L. Lewis, Zdenko Kovačić and José Mireles Jr.

Windup in Control
Peter Hippe

Nonlinear H_2/H_∞ Constrained Feedback Control
Murad Abu-Khalaf, Jie Huang and Frank L. Lewis

Practical Grey-box Process Identification
Torsten Bohlin

Control of Traffic Systems in Buildings
Sandor Markon, Hajime Kita, Hiroshi Kise and Thomas Bartz-Beielstein

Wind Turbine Control Systems
Fernando D. Bianchi, Hernán De Battista and Ricardo J. Mantz

Advanced Fuzzy Logic Technologies in Industrial Applications
Ying Bai, Hanqi Zhuang and Dali Wang (Eds.)

Practical PID Control
Antonio Visioli

(*continued after Index*)

Chunlei Zhang • Raúl Ordóñez

Extremum-Seeking Control and Applications

A Numerical Optimization-Based Approach

Springer

Chunlei Zhang
Applied Materials, Inc.
E. Arques Ave. 974
Sunnyvale, CA 94085
USA
zhangchunlei@gmail.com

Raúl Ordóñez
Dept. Electrical & Computer Engineering
University of Dayton
College Park 300
Dayton, OH 45469-0232
USA
ordonez@ieee.org

ISSN 1430-9491 Advances in Industrial Control
ISBN 978-1-4471-2223-4 e-ISBN 978-1-4471-2224-1
DOI 10.1007/978-1-4471-2224-1
Springer London Dordrecht Heidelberg New York

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2011941276

Cover design: VTeX UAB, Lithuania

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To Ran (C.Z.)
To Junko and Kaito (R.O.)

Series Editors' Foreword

The series *Advances in Industrial Control* aims to report and encourage technology transfer in control engineering. The rapid development of control technology has an impact on all areas of the control discipline. New theory, new controllers, actuators, sensors, new industrial processes, computer methods, new applications, new philosophies..., new challenges. Much of this development work resides in industrial reports, feasibility study papers and the reports of advanced collaborative projects. The series offers an opportunity for researchers to present an extended exposition of such new work in all aspects of industrial control for wider and rapid dissemination.

A monograph that was very influential in the field of extremum-seeking control was that due to K.B. Ariyur and M. Krstić entitled *Real-Time Optimization by Extremum-Seeking Control* published in 2003. This was a very timely contribution to the literature since it created a focus for the current research activity into the theory and application of extremum-seeking control, an activity that has continued to grow in the intervening years.

Many industrial and manufacturing processes are often "nudged" by operators into operating conditions that give better outcomes, productivity, and yields. Subsequently the controller design may need some adjustment to take advantage of the process features in the new operating condition. Extremum-seeking control is one formal way of finding this optimised process location. However, unlike optimal control design methods per se, in extremum-seeking control the performance cost function is often unknown analytically, and the optimization has to proceed using online real-time measurements of actual or related process and system variables. Consequently, it is not surprising to find that extremum-seeking control has drawn on the wealth of research into numerical optimization methods.

This *Advances in Industrial Control* series monograph *Extremum-Seeking Control and Applications* by Chunlei Zhang and Raúl Ordóñez has the subtitle *A Numerical Optimization-Based Approach* that clearly indicates the constructive numerical optimization route the authors intend to follow. The monograph has an introductory overview chapter and then the remaining chapters fall into two parts, theory first followed by the application studies. There are four chapters of theory. Chapter 2 reports the numerical optimization results and techniques the authors will be using. This is followed by the extremum-seeking control design chapter and then two chapters on the development and analysis for the numerical-optimization based extremum-seeking control algorithm devised by the authors. The results in these chapters are illustrated by examples and simulation studies.

In the all-important applications part of the monograph, the reader will find detailed expositions of three applications, each presented at chapter length in Chaps. 6, 7, and 8 respectively. Usefully, these chapters present a benchmark process, a real-world industrial application, and a potential field of future applications. Somehow, the antilock brake system almost seems to have become a benchmark process for demonstrating extremum-seeking control methods and this is the topic of Chap. 6. The real-world industrial problem of Chap. 7 involves extremum seeking control design and implementation for a semiconductor plasma-processing chamber, and in Chap. 8, the future application is the co-ordinated control of a swarm of agents (particles) where the solution of a source seeking and surround problem is investigated. These chapters have the merit of presenting a tutorial application, a report of what happens in a real industrial application of extremum-seeking control and a look at a more futuristic application that appears frequently in quite a number of today's international control research conferences.

This monograph will be of interest to readers from a wide range of disciplines, from applied industrial mathematics, through the mechanical, electrical, manufacturing and chemical engineering disciplines and on to the control systems and control engineering fields. The level of the presentation is appropriate for industrial engineers and researchers, postgraduate students and the control academic research and practitioner communities. The Editors are pleased to welcome this very first monograph on extremum-seeking control in the *Advances in Industrial Control* series.

Industrial Control Centre
Glasgow, Scotland, UK

M.J. Grimble
M.A. Johnson

Preface

There are many problems of engineering interest for which an optimal operating point or condition exists, but this point or condition are not necessarily well known or easy to find. Extremum seeking control is a family of control design methods whose purpose is to autonomously find an optimal system behavior (e.g., set point or trajectory to be tracked) for the closed-loop system, while at the same time maintaining stability and boundedness of signals. Extremum seeking control is therefore mainly used to realize real-time optimization for dynamic systems. It has been applied to engineering problems in the automotive industry, process control, thermal fluids, flow control, semiconductor industry, energy conversion and many other areas. The motivation for the research on extremum seeking control arises from its practical interest, since even small improvements in performance can lead to cost and energy savings. This book reviews existing extremum seeking techniques, and proposes a new numerical optimization based extremum seeking control approach. Several applications are presented, including problems from the automotive industry, autonomous robotics and the semiconductor industry. This book will be of benefit to students and professionals in all areas of engineering, especially on system control and optimization. The book contains many figures (block diagrams, plots, simulation and experimental results) that will help the reader to understand the material. The reader (researcher, Ph.D or M.S level graduate student, R&D engineer) will become familiar with step-by-step algorithms that can be readily applied to a variety of control problems in engineering practice. The reader will also acquire a deep understanding of extremum seeking control and its mathematical foundations. The application examples included in the book will help the reader understand the concepts and how they can be applied.

Overview of the Book

Tracking a varying maximum or minimum of a performance (output, cost) function is called extremum seeking control. For example, problems where finding such an extremum may be of interest include maximizing the yield of bioreactors, minimizing the power demand in formation control, minimizing reflected power, or even producing a better tuning of the PID coefficients in a control system. All of these are problems of practical interest in engineering, some of which have ad-hoc solutions, and some of which are yet unresolved. It is for these types of problems that extremum seeking control can be of great value. Extremum seeking attempts to determine the optimal performance of a control system as it operates, thereby potentially eliminating or reducing the need for down time and for doing system analysis.

In the first part of the book, we begin with a comprehensive review of the state-of-the-art in the extremum seeking control literature, and provide the reader with an understanding of what the different "flavors" of extremum seeking control are, and how they relate to each other. We review the existing analog optimization based extremum seeking control methods, which include gradient based design, perturbation based design and sliding mode based design. Then, we present a novel numerical optimization based extremum seeking control method that makes use of numerical optimization algorithms and state regulation, starting from simple linear time invariant systems and extending to a class of feedback linearizable nonlinear systems. We also analyze the robustness of two main optimization algorithms as they apply to extremum seeking: line search methods and trust region methods. For linear systems, a finite time state regulator is proposed and an asymptotic state regulator is used for nonlinear systems. Further design flexibility is achieved via the robustness results of the optimization algorithms and the asymptotic state regulator, where existing nonlinear adaptive control techniques can be introduced for robust design.

The second part of the book deals with the application aspects of extremum seeking control. We perform a comparative study of antilock braking system design via different extremum seeking control schemes. An industrial application of extremum seeking control methods to RF impedance matching is also presented, including experimental results obtained by one of the authors (Zhang) at Applied Materials. Finally, an interesting and promising application studied here is the autonomous agent source seeking problem. We use extremum seeking control and artificial potentials to achieve source seeking, formation control, collision avoidance and obstacle avoidance of a group of autonomous agents. Within this context, we present a practical application of source seeking via extremum seeking control to mobile radar sensor networks.

Acknowledgements

The authors gratefully acknowledge Professor Paul Eloe (University of Dayton), Professor John Dennis (Rice University) and Professor Mark A. Abramson (Air Force Institute of Technology) for their help on numerical optimization and analysis (Chap. 2); Professor Ümit Özgüner (Ohio State University) for discussions on sliding mode based extremum seeking control (Chap. 3); Professor Miroslav Krstić, Daniel Arnold, Antranik Siranosian and Nima Ghods (all at University of

California at San Diego) for their collaborative research on perturbation based extremum seeking control (Chap. 3); Professor Ruihua Liu (University of Dayton) and Professor Manfredi Maggiore (University of Toronto) for their valuable suggestions for NOESC on LTI systems (Chap. 4); Kartik Ramaswamy, Jim Cruse, Sergio Shoji, Lawrence Wong, Bryan Liao, Andrey Semenin and Hiroji Hanawa (all at Applied Materials) for the work of impedance matching (Chap. 7); Professor Kevin M. Passino (Ohio State University) and Professor Veysel Gazi (Istanbul Kemerburgaz University) for their pioneering work on swarm source seeking (Chap. 8); Jingyi Yao for her collaboration on swarm source seeking problem (Chap. 8); and Dr. Atindra Mitra and Sean Young (Air Force Research Laboratory) for their help in the formulation of a potential function in the work on radar leakage point localization (Chap. 8).

This work would not have been possible without the help from Professor Miroslav Krstić (University of California at San Diego) and Professor Qin Sheng (Baylor University). One of the authors (Zhang) benefitted significantly from the guidance on extremum seeking control from Professor Miroslav Krstić, on numerical optimization from Professor Qin Sheng, and from their dedication to academic research.

The authors also like to thank Kayode Ajayi-Majebi for the Khepera experimentations using extremum seeking control; Jia Wang for the potential fields and decentralized control; and Hai Yu for his insightful communications on extremum seeking control.

C. Zhang would like to thank his colleagues Kartik Ramaswamy, Sergio Shoji, Jim Cruse, Shahid Rauf and Ken Collins at Applied Materials for their great support and all they have taught him about the semiconductor industry.

R. Ordóñez would like to thank his PhD dissertation adviser, Professor Kevin M. Passino (Ohio State University), for his life-long dedication to scholarship, and for his humanist approach to teaching and research, all of which have been a constant inspiration and a source of guidance.

Finally, we would most like to thank our families, who have been lovingly supportive through all the time of intense work devoted to this book.

Santa Clara, CA, USA Chunlei Zhang
Dayton, OH, USA Raúl Ordóñez

Contents

Part II Applications

Acronyms

ABS	Antilock Braking Systems
AOESC	Analog Optimization-based Extremum Seeking Control
APC	Advanced Process Control
CCP	Capacitive Coupled Plasma
CFD	Computational Fluid Dynamics
CVD	Chemical Vapor Deposition
DC	Direct Current
ESC	Extremum Seeking Control
EGR	Exhaust Gas Recirculation
FEKO	Electromagnetic simulation software manufactured by EM Software & Systems-S.A.
FTU	Frascati Tokamak Upgrade
GESC	Gradient-based Extremum Seeking Control
GPS	Global Positioning System
HCCI	Homogeneous-charge-compression-ignition
HF	High Frequency
HVAC	Heating, Ventilation and Air Conditioning
IPMC	Ionic Polymer Metal Composite
LC	A Circuit with an Inductor and a Capacitor
LF	Low Frequency
LQR	Linear Quadratic Regulator
LTI	Linear Time Invariant
MEMS	Microelectromechanical Systems
MPC	Model Predictive Control
MPPT	Maximum Power Point Tracking
NFL	No Free Lunch
NOESC	Numerical Optimization-based Extremum Seeking Control
OPA	Optical Phased Arrays
PEC	Perfect Electric Conductors
PDE	Partial Differential Equations
PESC	Perturbation-based Extremum Seeking Control

PID	Proportional-integral-derivative
PO	Physical Optics code
RF	Radio Frequency
SIR	Signal-to-interference ratio
SISO	Single Input, Single Output
SMESC	Sliding Mode-based Extremum Seeking Control
SPESC	Sinusoidal Perturbation-based Extremum Seeking Control
SPSA	Simultaneous perturbation stochastic approximation
SUAV	Small Unmanned Air Vehicle
TOC	Traditional Optimal Control
UAV	Unmanned Air Vehicle
UTRC	United Technologies Research Center
VSWR	Voltage Standing Wave Ratio

Part I
Theory

Chapter 1
Introduction

1.1 Motivation

Traditional control system design deals with the problem of *stabilization* of a system about a known reference trajectory or set point, while attaining certain design criteria. This general stabilization problem includes the so-called "tracking" and "regulation" problems, where the reference is often easily determined or generated. However, in some occasions it can be quite difficult to find a suitable reference value. For instance, the fuel consumption of a car depends on the ignition angle. If one desires to maintain optimal efficiency, it is necessary to change the ignition angle as the conditions of the road and the load of the car change.

Tracking a varying maximum or minimum (also referred to as extremum, or optimum value) of a performance function (herein also referred to as output, or cost function) is called extremum seeking control [4]. In order to better explain the concept of extremum seeking control and its motivation, next we discuss two relevant examples from industrial control engineering practice.

1.1.1 Anti-lock Braking Systems

Nowadays, nearly every new automobile is equipped with Anti-lock Braking Systems (ABS) [1–3]. ABS is a safety system whose tuning of the coefficients in purpose is to achieve the maximum possible braking force in a motor vehicle's wheels while preventing the wheels from locking up, and therefore avoiding skidding. There exist a PID number of approaches for the design of ABS, and here we consider ABS design using extremum seeking control.

In an extremum seeking control implementation of ABS, the controller automatically applies the brake such that the friction force coefficient between the wheel and the ground is maximized, and therefore the vehicle is stopped in a minimum time, where the torque control is realized via extremum seeking. Figure 1.1 presents a simplified block diagram of ABS design via extremum seeking. The objective of

C. Zhang, R. Ordóñez, *Extremum-Seeking Control and Applications*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2224-1_1, © Springer-Verlag London Limited 2012

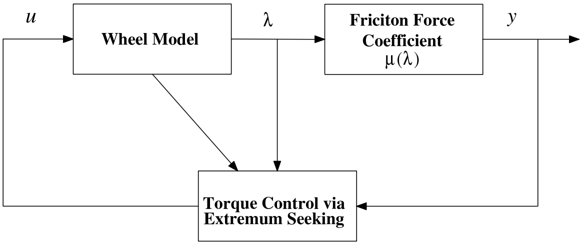

u
Wheel Model
λ
Friciton Force Coefficient μ(λ)
y
Torque Control via Extremum Seeking

Fig. 1.1 Anti-lock Braking Systems

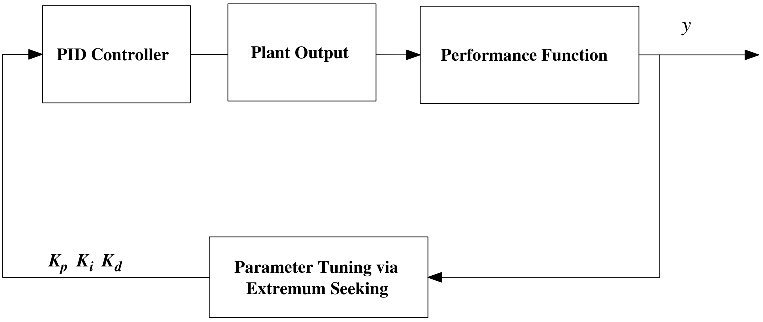


Fig. 1.2 PID Tuning

the ABS extremum seeking controller is to maximize the friction force coefficient μ, which has a maximum at some non-zero value of slip λ. The problem is complicated by the fact that the friction coefficient is unknown and cannot be directly measured. Moreover, it changes for different road and wheel conditions. Thus, some sort of indirect measurement or estimation of this friction force coefficient needs to be fed back to the extremum seeking controller. Chapter 6 gives a detailed description of the ABS problem and presents the reader solutions based on different extremum seeking control methods.

1.1.2 PID Tuning

Proportional-integral-derivative (PID) controllers are widely used in industry. At the same time, the tuning of PID parameters is often done in manually, and thus the effectiveness of the tuning is typically based on the experiences of the control engineer. Furthermore, in many practical situations, the plant model is unknown or only approximately known. Thus, a method for tuning PID controller in the closed-loop setting is very appealing. Extremum seeking control is very well suited for performing automated PID tuning [8]. Figure 1.2 shows a schematic block diagram of the tuning system. A cost function is composed based on the difference between the actual response and the desired response, and this cost function output is then

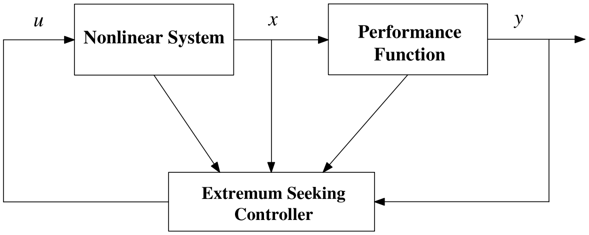


Fig. 1.3 General block diagram for extremum seeking control

fed back to the extremum seeking loop in order to tune the three parameters K_p, K_i and K_d of the PID controller.

There exist a large number of additional relevant examples of industrial applications of extremum seeking control, and these will be covered in more detail in Sect. 1.3.

1.2 Basic Principles

A general block diagram of extremum seeking control can be found on Fig. 1.3. Extremum seeking control has two functional layers: first, we need to seek an extremum of the output function;(or performance) function; secondly, we need to be able to control (stabilize) the system and drive the performance output to that extremum.

It is important to place extremum seeking control (ESC) in the context of the systems and control literature at large. In doing so, one notes that ESC is highly related to both traditional optimal control (TOC) [9–11] and model predictive control (MPC) [12, 13]. In the traditional optimal control problem, the performance function generally is an integral function of the state and control. That is, the extremum of the performance function is a trajectory. Therefore, calculus of variations is involved in the design, and the analytical form of the performance function is needed to obtain the necessary conditions of the optimal control. Related to optimal control, model predictive control (MPC) is one of the most widely applied advanced control methods in industry, where the control input at each step is obtained by solving an optimization problem. In this case, the optimization objective is to try to approximate an optimal trajectory in the context of optimal control using numerical optimization algorithms.

The common thread between the three control approaches (TOC, MPC and ESC) is that they all try to solve the problem

$$\min J(.) \quad \text{subject to } \dot{x} = f(x,u), \tag{1}$$

where J is a cost function that may depend on states, control input, time, initial and final values, etc.

In order to solve this problem, both TOC and MPC methods generally assume that the function (or functional) J is perfectly known—in fact, it is usually a design choice. However, in extremum seeking control,[1] the performance function is generally a function of the state and is unknown or poorly known in the design—and it is often a feature of the control problem itself. This lack of knowledge implies that we can only design an extremum seeking controller based on the *measurements* of the performance function or its derivatives, if available. In ABS design, for example, the friction force coefficient is a function of the slip, but this function is not known, and it also varies with different road conditions and weather. However, we can get measurements of the friction force coefficient, which can act as a proxy for the performance function we wish to optimize (the slip). Another example can be found in the problem of minimizing the effect of an unknown vortex field in formation flight [15], where again certain sensor measurements can be used as proxies of the effect of the unknown vortex field. At the same time, it is important to note that even if the performance function were perfectly known, and we could in principle get optimal points by finding the roots of its gradient, this may be hard to do in itself. Mathematically, the root finding and the optimization problems are equivalently difficult to solve.

The assumption about knowledge of the cost J is a fundamental difference between ESC and TOC and MPC, not only in practice, but also in the theory. This is because the availability of this knowledge, or lack thereof, puts constraints on what can be done with the system.[2]

In addition to the assumptions about knowledge of the cost function, model predictive control requires the assumption that it has a very good model of the system, good enough that it can act as a predictor. In this sense, MPC is very much like linear optimal control (TOC): the problem it actually solves can be loosely stated as

$$u(t) = \arg\big\{\min J(.) \text{ subject to } \dot{x} = f(x,u), \text{ for } t \in [t_k, t_{k+1}]\big\}.$$

In TOC, one may have $t_0 = 0$ and $t_1 = \infty$, as happens for the development of the linear quadratic regulator (LQR) [16] . In general, though, both MPC and TOC yield an *open loop* solution, that is, a signal $u(t)$ valid over some interval. Other than the LQR, which can be also expressed as a closed-loop solution, TOC generally lacks robustness. This is not necessarily true of MPC, since here the interval $[t_k, t_{k+1}]$ is usually small, and as the discrete index k increases, the signal $u(t)$ is recomputed. In practice, however, this can usually mean a computationally expensive method.

At the same time, the version of ESC presented in this book relies on an existing, possibly robust, closed-loop regulator or tracking controller, to which a sequence of set-points is passed. Thus, one can see that ESC does not quite solve the same

[1] Some results in the ESC literature do make the assumption that J is known (see [14], for instance). Our approach here is to avoid having to make this assumption, since otherwise the practical value of the method is somewhat limited, in our view.

[2] Philosophically, it may be useful to compare this difference to solving the regulation (or tracking) problem using state feedback versus output feedback. Both try to achieve the same ultimate objective, but the assumptions made about the available information profoundly alter the way one goes about achieving it.

problem as MPC and TOC. Moreover, it is important to emphasize that each method has its strengths and weaknesses, and its own domain of applicability.

Without assuming knowledge of the performance function, many of the ideas used in extremum seeking control have been transferred from numerical optimization, where only the measurements of the performance output or its gradient are required. Extremum seeking control is therefore also sometimes called dynamic optimization, due to the fact that the arguments of the performance function to be optimized are constrained by a dynamic system [17]. Also, the controller needs to track the extremum if it is varying. Control of air-fuel ratio of combustion is one example where the extremum will change with the temperature and fuel quality. Also, the maximum of tire friction force coefficient will change with temperature and road condition. Similar statements of extremum seeking control can be found in the literature. Whenever off-line calculation of the optimal parameters is impractical or when no reliable model is available to predict the variation of the cost function with time, extremum seeking algorithms are used to determine and track the parameters that optimize a system level cost function in real time [18].

The goal of extremum seeking control is to operate at a set-point that represents the optimal value of a function being optimized in the control loop [19]. It is also known as extremum control and self-optimizing control, meaning a control system that is used to determine and to maintain the extreme value of a function [20]. As stated in [3], extremum seeking control is applicable in situations where there is a nonlinearity in the control problem, and the nonlinearity has a local minimum or a maximum. The nonlinearity may be in the plant, as a physical nonlinearity, possibly manifesting itself through an equilibrium, or it may be in the control objective, added to the system through a cost functional of an optimization problem. Hence, one can use extremum seeking control both for tuning a set-point to achieve an optimal value of the output, or for tuning the parameters of a feedback law.

There are some aspects in extremum seeking control that need further investigation, beyond what is presented in this book. These include the modeling of the processes and nonlinearity [4]; taking into account transient dynamical phenomena; performance measurements with noise, where low-pass filtering the output can take care of the measurement noise but can also introduce additional dynamics into the problem [18].

In Sect. 13.3 in Aström and Wittenmark [4], the authors classify extremum seeking control among the most promising future areas for adaptive control. Extremum seeking control is of great practical interest, since even small improvements in the performance can lead to large savings in raw material and energy consumption. There are commercial extremum seeking controllers and applications, some of which will be presented and explored in this book.

1.3 Literature Overview

The emergence of extremum seeking control dates as far back as the 1922 paper of Leblanc [21], and it was already popular in the 1950s and 1960s. A historical

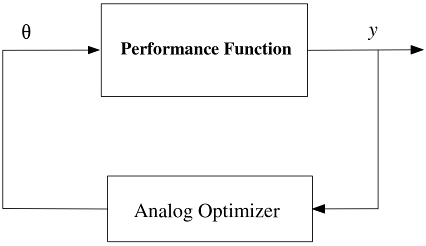


Fig. 1.4 Static Optimization

development of extremum seeking control can be found in [3]. Extremum seeking control witnessed a resurgence of interest after Krstić's publication of stability studies on perturbation-based extremum seeking control (PESC) in [23] and [24]. From a recent overview on, extremum seeking from 1922 to 2010 [25], one can find out that the number of publications increased by roughly a factor of eight during the first decade since the year 2000, compared with the previous ten years.

1.3.1 Theory Development of Extremum Seeking Control

By posing different assumptions on the knowledge of the system, performance output and so on as seen in Fig. 1.3, one ends up with various approaches for extremum seeking control design in the literature. Here, we mainly focus on the development through the last decades.

1.3.1.1 Static Optimization

In the traditional application of extremum seeking control, the optimal parameters were assumed to vary rather slowly. Then, system dynamics were typically neglected and the algorithm was analyzed and designed using traditional "static optimization" techniques [18, 22]. Many investigations of extremum seeking control assume that the dynamic system is static, which can be justified if the time between the changes in the optimal reference is sufficiently long. Extremum seeking control of static systems is in essence a problem of numerical optimization [4], which is generally solved iteratively in a discrete-time fashion. The gradient-based and nearest-neighboring extremal scheme is studied in [62] and output feedback is used to simplify the requirement of using state feedback for the update law. However, this static optimization can be approached using the continuous (analog) implementations of some numerical optimization methods; that is, the so-called "analog optimization" approach (such as sinusoidal perturbation, sliding mode-based analog optimization and gradient feedback). These analog optimization methods provide an important basis for extremum seeking control design.

1.3.1.2 Analog Optimization Based Extremum Seeking Control

By posing certain assumptions on a nonlinear system [3], one can reduce the extremum seeking control problem to a one dimensional optimization problem. There-

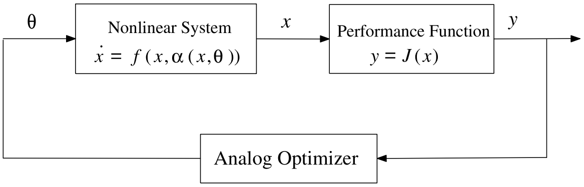


Fig. 1.5 Analog optimization based extremum seeking control

fore, the design of extremum seeking control focuses on how to find an updating law for the single parameter, θ, where several interesting analog analog optimizers [23, 26, 27] come into the context of extremum seeking control (refer to Fig. 1.5). The proper assumptions to allow time scale separation between the nonlinear system dynamics and the extremum seeking loop are crucial for analog optimization based extremum seeking control (AOESC) to be successful. Mathematical tools such as singular perturbation, averaging and Lyapunov functions are used for the stability analysis. Sinusoidal perturbation based optimization method is the most popular method used in extremum seeking control design. A high-pass filter and a slow sinusoidal perturbation signal are employed to derive the gradient information from the performance output. The first analysis of local stability of PESC for a general nonlinear system was developed based on averaging analysis and singular perturbation [23]. The pioneering averaging studies of Meerkov [28–30] stand out as a precursor to the stability results in [23]. In [24], dynamic compensation was proposed for providing stability guarantees and fast tracking of changes in the plant operating conditions for single parameter extremum seeking. Discrete time PESC was studied in [31]. An accelerator designed based on polynomial identification and Butterworth filter is studied in [32] to speed up the convergence of PESC. Roeta [18] and Walsh [33] provided the first studies of a multiparameter PESC scheme. Their results were for plants with constant parameters and a systematic design procedure is absent. Stability analysis for general multiparameter PESC and systematic design guidelines for stability/performance are supplied in [34]. The results in [35] constitute a generalization of extremum seeking, which seeks a point of zero slope, to the problem of seeking a general slope. The book by Ariyur and Krstić [3] presents a systematic description of PESC and its applications.

A simple PESC scheme is presented in [36], where also the semi-global stability analysis is discussed. These results then get extended to seek an optimal trajectory in a finite time interval in a repeatable control environment [37]. The study of different perturbation signals other than sinusoidal ones appeared in [38], where mathematical analysis and simulation results illustrate that mainly the shape of the perturbation signal will impact the performance of the extremum seeking control in terms of accuracy, speed and domain of convergence. Global extremum seeking is studied in [39] by changing the amplitude of the perturbation signal to overcome local minima, i.e., a larger perturbation amplitude is more likely to explore a wider range. The PESC above relies on time scale decomposition and as such has so far been developed only for plants that are open loop stable, with poles that are sufficiently well damped. A new idea based on phase lead compensator extends the applicability to moderately unstable systems [40]. For the multiparameter PESC case, the orthogonality requirement for the periodic perturbation signal presents an additional challenge on implementations (such as making the search vehicle's behavior more predictable in source seeking applications), and the periodicity of perturbation may cause unwanted resonant behavior in some systems. A stochastic perturbation signal is therefore studied in [41, 42] as a potential solution for these problems, which is also inspired from biological systems (such as bacterial chemotaxis) [43]. In the recent study [44], a Newton-like step is added to SPESC to further improve the effectiveness of extremum seeking by reducing the dependency on objective function curvature.

Sliding mode control is also a tool that has been used for optimization [45]. Similar to the perturbation method, a sinusoidal signal is used to generate the switching function to guide the controller seeking a different direction (using the sign of the gradient information) to find the extremum. Sliding mode based extremum seeking control (SMESC) is introduced by Korovin and Utkin in [46], and analyzed and applied by Özgüner and his coworkers [26, 47–55], and others [56–58] on a variety of automotive problems, especially in ABS design [1, 59–61]. The Lyapunov method is used for the analysis of convergence and stability of the closed-loop system.

Gradient (or its estimation) based extremum seeking control (GESC) is the most straightforward approach. A series of work has been carried outpapers by Banavar and his co-workers [19, 63–66] are strongly reminiscent of steepest descent type algorithms in numerical optimization. Continuing work has been seen on using Kalman filter to estimate the gradient and Hessian of the performance function in [67]. A similar idea also appears in the parameter updating law design in adaptive control [68]. The self-optimizing control studied by Horowitz and his co-workers [69–71] belongs to estimated gradient based ESC. Recent developments of gradient based ESC with adaptive design are pursued by Guay and his co-workers. An ESC problem is proposed and solved in [14] for a class of nonlinear systems with unknown parameters, where an explicit structure information for the performance function to be maximized is required. Assuming that one can provide a suitable functional expression for the plant profit, an adaptive receding horizon controller design technique is developed that is able to steer the process state of the closed-loop system to an unknown optimum while ensuring transient performance and process regulation about the unknown optimum [72]. They continue the work for output feedback ESC for linear uncertain plants in [73], state constrained nonlinear systems in [74] and a flatness based approach for nonlinear systems in [75]. Further study on persistently exciting signals used to guarantee parameter convergence appeared in [76].

The optimization process involved in AOESC is continuous, as opposed to general iterative numerical optimization algorithms. The requirement for continuous measurements of the gradient is very strong, thus PESC and SMESC are appealing because they are non-gradient based. Other non-gradient analog optimization such as neural-network based can be found in [27, 77]. The main advantage claimed by AOESC methods is non-model based design due to the assumption of asymptotic system stability, and the analog optimization for updating the parameter does not require the plant knowledge. However, for more complicated systems involving

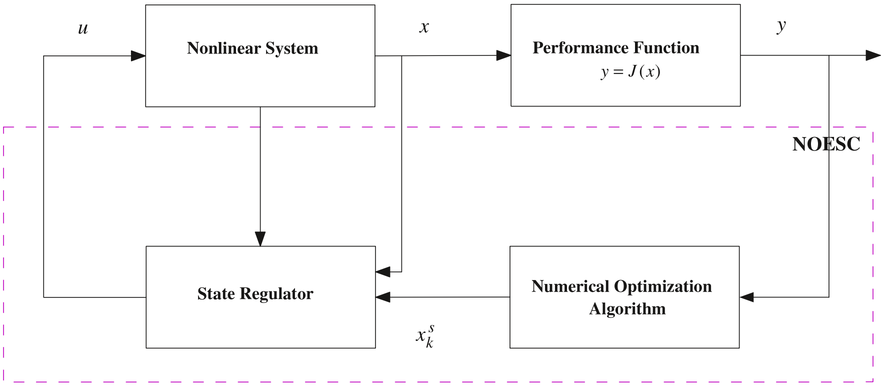


Fig. 1.6 Numerical optimization based extremum seeking control

multiple parameters (such as the source seeking example in Chap. 3), the system model is in fact needed for the design of extremum seeking controller using AOESC methods. Recent analysis on general AOESC can be found in [78, 79].

1.3.1.3 Numerical Optimization Based Extremum Seeking Control

Numerical optimization-based extremum seeking control (NOESC) methods have been in development somewhat separately from AOESC approaches (refer to Fig. 1.6). Now, the requirement for gradient measurements is not continuous, and the system may have time to collect enough output measurements to estimate the gradient. Moreover, there are non-gradient numerical optimization algorithms that can be implemented as well. Extremum seeking control via triangular search as in [80] was employed to attenuate combustor thermoacoustic oscillations and minimize diffuser losses at United Technologies Research Center (UTRC). Nonlinear programming was successfully used in ESC by defining a readout map as a steady-state output function [81]. Simultaneous perturbation stochastic approximation (SPSA) recursive algorithm is used in the ESC design [17] and [41]. More systematic studies by Zhang and Ordóñez on NOESC first appeared in [82], and then in [83–85], where numerical optimization algorithms and state regulation are combined to design the extremum seeking control scheme, and will be presented in this book. The extremum seeking controller (state regulator) ensures that the state travels along the set-point sequence generated by the numerical optimization algorithm, which eventually converges to a minimizer of the performance function. NOESC design does require a system model to construct a state regulator to achieve the dynamic optimization; however, the design method is more generic (it is a multiparameter ESC design and is not limited to any particular optimization algorithm). Therefore, it can be applied to different systems and improve the control performance by choosing a suitable optimization algorithm. Two different kinds of state regulator design, finite-time state regulator and asymptotic state regulator, are studied in this book. Moreover, if the state regulator design does not require a system model (for instance, by making asymptotic stability assumptions similar to those in PESC schemes), the NOESC method could be considered a non-model based design as well. A similar design framework is studied with particle swarming optimization [86] and bacteria foraging optimization [87]. State regulation is achieved by predictive control for LTI systems in [88] as the set-point sequence becomes the terminal state constraint of predictive control. Furthermore, instead of using a state regulator, a tracking control is used to achieve extremum seeking given that a reference trajectory is obtained by filtering the set-point sequence [89].

1.3.1.4 Other Approaches

Many other research work in ESC arises from different practical motivations and assumptions. An extremum controller having no special trial steps or oscillations can be designed if some a priori knowledge about the plant and its disturbance exist [90]. That is, the extremum characteristic of the plant is unimodal and can be approximated by second or third-order polynomials, the distribution of the disturbances is close to uniform and the value of the optimum changes slowly. A self-tuning concept is applied to the extremum control problem in [91]. It can be viewed as the case of online optimization of a static performance criterion. When placed in a self-tuning context, this performance index will be parameterized as a quadratic function so that a recursive estimator can be applied to determine the coefficients of its parametric form. The mixing of dynamics with the static optimization task presents problems for the self-tuning identifier [92], which is a companion paper concerning the application of self-tuning extremum control to automotive engine management. Adaptive extremum control [91, 93–95] is intended to optimize the output of static nonlinear processes under noisy conditions. The concept of explicitly parameterizing the performance index and estimating the coefficients has appeared in [96]. Extremum seeking control dealing with the steady-state optimization of noisy nonlinear systems is studied in [95]. The control action is calculated by making use of a model obtained by some kind of system identification. A new approach called Dynamic Extremum Control was studied in [97, 98] and earlier references of the authors. The authors explicitly solve the optimal condition given the form of the cost function, and a self-tuning algorithm is then developed, where a dither signal is used to prevent non-identifiability problems.

Extremum seeking control of Wiener type systems is considered in [99], where the linear subsystem is described by a discrete-time system with delay and Gaussian distributed white noise. The nonlinearity is described as a quadratic function with a unique minimum. Therefore, the author solves for the minimum of the output and the purpose of the control is to keep the output as close as possible to the minimum. Extremum seeking control based on a probing strategy can be found in [100], and the stability and performance issues are examined further in [101]. Discrete-time extremum seeking algorithms for SISO nonlinear systems are proposed in [102], where the reference-to-output equilibrium map is approximated via a quadratic polynomial. The extremum can be explicitly solved by the polynomial

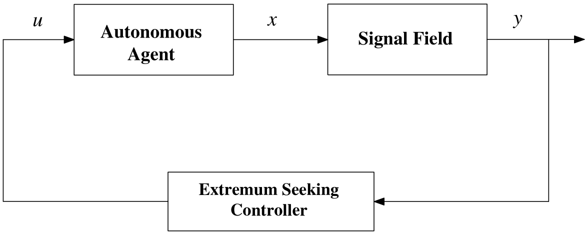


Fig. 1.7 Source Seeking

parameters, then three extremum seeking algorithms are proposed based on the way they estimate the polynomial parameters, which are least square estimation, parabola approximation and ellipse approximation. A similar study can be found in [103].

1.3.2 Applications of Extremum Seeking Control

Among the many applications of extremum seeking control overviewed in [22] and [4] are combustion processes, grinding processes, solar cell management, radio telescope antenna adjustment to maximize the received signal, and blade adjustment in water turbines and wind mills to maximize the generated power. Even though most of the application results are presented as simulations in the literature, there are effective experimentation results that have been obtained in real engineering problems [44, 104–115]. Both simulation and experiment applications on theory and engineering problems are reviewed in this section.

1.3.2.1 Agents and Sensor Networks

Control of a single or a group of autonomous agent and sensor networks is one of the most active research areas in the last decade. Source seeking (or source localization), i.e., the design of control algorithms for autonomous agents to seek a source with unknown spatial distribution is of great interest, where extremum seeking control can be naturally used in the design. Even though it is an application having great theoretical interest, it also does have a significant impact on engineering applications: for instance, in the problem of developing vehicles with more autonomy, such as the situation where no GPS information is available, or to reduce cost due to position sensors. Some of the direct applications of source seeking can be found in contaminant plume control, autonomous odor sensing or toxic gas leakage localization. In the application of source seeking, the task of the vehicle is to find a source that emits a signal that decays as a function of distance away from the source, where the signal field is unknown and only the measurement of the signal at the current agent location is available.

A basic diagram of source seeking can be found in Fig. 1.7, where the control goal is for the agent to seek an extremum of an unknown signal field based on the measurement of the signal only. Source seeking is first discussed as a direct application result of PESC for systems with moderately unstable single poles [116], where the autonomous vehicle is modeled as a single or a double integrator. A formal study on unicycle model source seeking first appeared in [117], where the design keeps the angular velocity constant and tunes the forward speed by extremum seeking, which generates triangle or star pattern vehicle motions that drift toward the source. A more challenging analysis is performed on a different strategy in [118], where the approach is to keep the forward speed constant and tune the angular velocity by extremum seeking. The resulting motion sinusoidally converges to the source and settles in a ring around the source as the forward speed is constant. Based on these two strategies, adding a simple derivative-like feedback to the forward speed in the angular velocity tuned by extremum seeking loop [119] allows the vehicle to slow down as it gets closer to the source and converges closer to the source without giving up convergence speed. Source seeking for slow or drifting sensors is explored in [120]. Rather than sinusoidal perturbations, a stochastic source seeking control law is used to tune the forward velocity in [121, 122] and the angular velocity in [123].

An extension to 3D source seeking is studied in [124], where two design schemes are studied: One actuates both yaw and pitch velocity while keeping the forward velocity constant. The other actuates the roll velocity while keeping the forward and roll velocity constant. Source seeking for fish models is discussed in [125, 126] and an overview paper of source seeking can be found in [127]. Moreover, SMESC is also used for non-holonomic vehicle source seeking in [128, 129]. In this work, unlike the above series of papers based on PESC, sliding mode optimization is used to control the on-board antenna toward the right direction to the source, and then sliding mode control and MPC are used to regulate the vehicle velocity and direction of driving. This two-stage control is similar to numerical optimization based extremum seeking control, where an optimization algorithm provides the set-point sequence and a state regulator drives the state. An advanced numerical optimization algorithm is proposed for source seeking in [130] with the focus on convergence and robustness, while the vehicle dynamics are not considered in part of the design, and therefore open loop control is used to steer the vehicle.

The application of ESC in group source seeking (namely, swarm seeking) has emerged recently as well. Swarm seeking is approached via a leader-follower format in [131], and extremum seeking control is used to guide the leader vehicle to seek the source; then, a passivity framework is used to design the followers' coordination laws. The extremum seeking control algorithm is basically a type of NOESC: the vehicle dynamics are given by a simple integrator, and Newton's method is used to construct the set-point sequence. For this purpose, a specific design vehicle velocity profile is used to estimate the gradient and Hessian matrix by measuring the field signal at orthogonal directions. Biased random walk inspired by bacterial chemotaxis is also used in swarm seeking [43]. Similar work based on stochastic extremum seeking can be found in [132]. Diffusion based feedback control is combined with perturbation based extremum seeking in the study of 1D swarm seeking [133], where extremum seeking tries to place all the agents at the extremum while a diffusion mechanism aims to spread the agents evenly. Averaging theory for PDE's is used to prove that the vehicle density is highest around the source. A detailed discussion and comparison of swarm seeking can be found in Chap. 8.

PESC is used to optimize the throughput of a linked communication chain of mobile nodes in [134]. The mobility of the nodes is modeled as the standard bicycle kinematic model, where the multi-variable PESC is implemented to achieve the decentralized control. Other work investigating the use of extremum seeking for formation control includes [7, 15, 135, 136].

1.3.2.2 Automotive and Related Applications

Automotive engine performance has been difficult to improve as the number of parameters to optimize keeps increasing. The traditional engine mapping and calibration has become a very time consuming task, and instead extremum seeking control has been widely explored in this area. In [107, 137], NOESC is used to tune intake cam timing, exhaust cam timing and spark timings to improve the brake specific fuel consumption of a variable cam timing engine. Three different numerical optimization algorithms (simultaneous perturbation stochastic approximation algorithm, persistently exciting finite differences algorithm, and Box and Wilson steepest descent algorithm) are compared, where repeatable experiment results are obtained. Homogeneous-charge-compression-ignition (HCCI) engines have the benefits of high efficiency with low emission of NOx and particulates, and one design goal is to minimize their fuel consumption, which is indirectly controlled by the combustion-timing. The combustion-time is a function of gas temperature, pressure, composition and mixture homogeneity. In [110], the combustion-timing is controlled by some temperature-control valves, and the author first uses perturbation based extremum control to perturb the combustion-timing set-point to optimize the fuel consumption, then a faster extremum seeking loop is used to tune the PID parameter of the valve control. PESC is used to minimize the engine performance index via controlling the throttle angle [138]. In [139], a SMESC is implemented to modulate spark time in a slow-time scale to maximize the steady-state Exhaust Gas Recirculation (EGR) amount. Further studies of engine optimization appear in [140, 141].

Another application of ESC in automotive industry is Antilock Breaking Systems (ABS), where different sorts of extremum seeking control design have been applied. These include SMESC, PESC and NOESC [1–3, 84]. A detailed design of PESC has been presented in [142], where Lyapunov analysis is used to study the design along the lines of the results in [36]. A discussion on the tuning of extremum seeking controller is also provided, simulation studies showed ESC also has good degree of robustness. Chapter 6 studies several different ESC designs for ABS.

In the context of aircraft engine design, extremum seeking control has been applied to reduce the acoustic pressure oscillations in gas turbine engines [104], where a phase shifting controller is tuned by PESC, whereas [143] considers application of PESC to an axial-flow compressor. Other applications include control of combustion instability [144, 145] and minimizing nitrogen oxides emission [146].

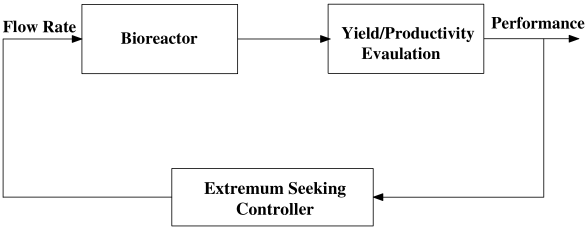


Fig. 1.8 Bioreactor

1.3.2.3 Process Control

The process industry is another area where control engineers have made much use of extremum seeking control. For continuous or fed-batch bio-processes, the productivity is in general an increasing function of input flow rate, while the yield is a decreasing function of the input flow rate. Operating close to 100% yield normally gives very low productivity, and vice versa. One design goal is to adjust the input flow rate to achieve the best trade-off between yield and productivity, i.e., optimize an objective function composed of both productivity and yield. The corresponding diagram can be found in Fig. 1.8.

Given the explicit knowledge of the growth kinetic model and objective function with some parameter uncertainty, while the actual measurement of the objective is not available (one justification is that such measurements could be expensive or inaccurate), adaptive extremum seeking control [14] is applied to bio-reactors with Monod growth kinetics in [6, 147], Haldane kinetics in [148] and fed-batch process in [149]. A numerical implementation for constrained optimization of a batch system can be found in [150]. The neural-network approximation technique is used to relax the requirement of growth kinetic model in [151] and [152].

A similar application for a tubular reactor with distributed feed can be found in [153–155], and the productivity optimization for a van de Vusse reaction is studied in [156]. Without knowing the nonlinear process dynamics and explicit form of the objective function, based only on the measurements of cost at a particular flow rate, the use of simple PESC [5, 36] is able to achieve the optimization as in [157], where the authors further explore the application to a multivalued objective function using a proper tuning of the perturbation signal amplitude to overcome local optima or bifurcation points. NOESC is applied to maximize the biomass production using Haldane and Monod growth kinetics [89]. Extremum seeking is applied to fermentation processes in [158].

1.3.2.4 Flow Control

In the area of flow control, PESC is used to minimize the global energy consumption (electric and aerodynamic power) of a flow system [159] via tuning the cylinder rotation speed. Extremum seeking [160] outperformed traditional look-up table methods or computational fluid dynamics (CFD) simulation, where the multiparameter PESC is successfully used to find the optimal torque and pitch angle based on the measurement of rotor power. Various design guidelines have been further discussed in [18]. In addition, anti-windup ESC is implemented via the usage of back calculation method, and resetting logic is embedded in the system to deal with a drifting extremum. An application of PESC to subsonic cavity flow control is presented in [161]. Related to this work, extremum seeking control of a high-speed and high Reynolds number jet is examined in [162].

For micro-channel based cooling of high-power electronic devices, gradient estimation based extremum seeking control is used to maximize the heat transfer coefficient, based on the measurements of the wall temperature [163]. Perturbation based extremum seeking control is used to enhance the heat transfer rate in magnetohydrodynamic channel flow [164, 165]. Other applications include flow separation control in diffusers [166] and control of an electromechanical valve actuator [167].

1.3.2.5 Plasma Control

In the plasma control area, one goal is to minimize the reflective radio-frequency (RF) power or to maximize the delivery RF power given different plasma loads. In the Frascati Tokamak Upgrade (FTU), the effectiveness of RF heating is proportional to the coupling between antenna and plasma, and extremum seeking control is used to maximize delivery power and therefore maximize the temperature increase given the same power set-point [168]. The implementation issues due to multiple antennas are studied in [106].

Gradient estimation based extremum seeking control is proposed for the plasma RF heating on FTU [169]. PESC is also used to minimize the reflective power by adjusting the plasma position through a build in position and plasma current control system in the Frascati Tokamak [108]. For safety concerns, instead of introducing the sinusoidal perturbation signal to the system, the existing power circuit disturbance is used as the perturbation signal for the extremum seeking loop. However, PESC is mainly used here as an optimization algorithm to adjust in real time the set-point for the plasma position control system, not directly adjusting the system control input. The work [170] deals with finite-time extremum seeking control of plasma in the Tokamak. RF impedance matching via extremum seeking for semiconductor plasma chambers will be discussed at length in Chap. 7.

1.3.2.6 Energy Conversion

In the fast-growing green energy area, maximum power point tracking (MPPT) control for energy conversion systems gained noticeable interest during the last few years. The common control input is the rotor speed of the generator, and the output is stator power (refer to Fig. 1.9). If one knew the optimal tip speed ratio and wind velocity, it becomes a set-point regulation problem. However, such measurements are difficult to obtain [171]. Then, based on the knowledge of the wind turbine's maximum power curve, MPPT is approached through trajectory tracking [172].

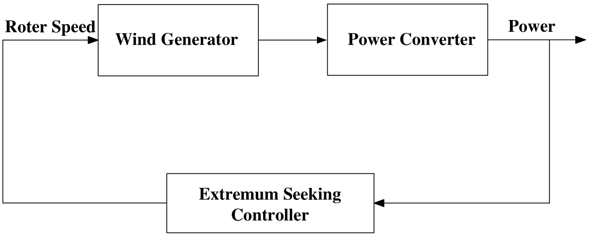


Fig. 1.9 Maximum power point tracking

However, the maximum power curve first needs to be estimated via simulation or experimentation; moreover, it fluctuates as the wind velocity changes. A simple idea from optimization, hill climbing, is used to design the MPPT controller [173]. Recent studies have used GESC [174], SMESC [175, 176] and PESC [113, 177–182] in the MPPT design. Constrained extremum seeking control for MPPT of a polymer electrolyte membrane fuel cell system is studied in [183], where a penalty function is used to transform inequality constraints to be part of the objective function.

Maximizing wind turbine energy capture via extremum seeking is briefly studied in [184]. In order to maximize the cooling power of the thermoacoustic cooler, multiparameter PESC is used to tune the driving frequency and piston position to maximize the heat transfer rate [105, 185]. Experimental results showed improved performance even in the presence of a changing flow rate. A Newton-like step is added to SPESC to minimize the thermoacoustic oscillations in a model premixed combustor [44]. Using the optical power and wavelengths of the optical pump lasers as control inputs [186], NOESC is used to minimize the variation of output amplifier optical signal power from a desired power over the entire operating signal wavelength band. For the beam matching problem, a multiparameter PESC is used to tune the lens strength in four-lens and six-lens channels to achieve matching [187–189]. In mobile wireless networks, sliding mode extremum seeking control is used to adjust the power at each node based on the feedback of signal-to-interference ratio (SIR), while maintaining the network equilibrium close to the Pareto-optimum [190]. Sinusoidal perturbation based extremum seeking control is used to improve the variator efficiency in a pushbelt continuously variable transmission [115].

1.3.2.7 Control Design and Optimization

One application of extremum seeking is in simplifying controller design. For an n dimensional system, $2n$ simple feedback control laws are designed that only use one state variable. A variable structure control algorithm [191] is proposed to stabilize the system by switching the system control input among the $2n$ feedback control laws, where the switching mechanism is achieved by SMESC by minimizing a Lyapunov function.

Potentially one of the most important applications of extremum seeking control for engineering problems in industry is the optimization of PID coefficients. The non-model based PESC [8] can be easily incorporated as an existing PID controller auto-tune feature, which will improve the controller performance without requiring advance knowledge for the user. This tuning method is applied to tuning a proportional term of a weight feeder controller in [114], and tuning PID controller for Tokamaks in [192]. Annealing recurrent neural-network based extremum seeking is also applied for PID tuning in [193]. Given the ease of use of auto-tune PID, one can improve the controller performance by simple firmware updates and therefore the cost benefits are significant as PID control is the most widely applied control method in industry.

Similar work on tuning backstepping controller parameters via SPESC can be found in [194]. Moreover, perturbation based extremum seeking control is also applied to game theory [51, 195–197] and limit cycle minimization [198]. Optimizing the nonlinear control of blending processes can be found in [199].

1.3.2.8 Other Industrial Applications

To conclude this overview of applications, we list some other recent work that does not easily fit the classification we have followed here.

Advances made in the research of Ionic Polymer Metal Composite (IPMC) actuators enables some application such as small scale aquatic vessels to operate within bloodstreams, in order to record and transmit biological data, or to interact with the blood to treat a medical condition. Both experimentation and simulation results are presented in [109], where an IPMC fish-tail and its controller swimming environment are manufactured, and PESC is used to maximize the swimming thrust via tuning the tail oscillation.

For MEMS vibrating gyroscopes, the mode-matching condition is essential in order to achieve enhanced sensitivity and uniform performance on the whole sensor bandwidth. The goal is to match the resonant frequencies of the sense-mode to the drive mode. This mode-matching problem can be reformulated as the problem of maximizing the sense-mode oscillation amplitude. The sense-mode natural frequency is adjusted via PESC to maximize the amplitude of the sense-mode dynamics at the drive-mode natural frequency in [200, 201].

In the application to drilling using minimal reaction force and torque while operating from light-weight platforms or planets with low gravity, an ultrasonic/sonic driller/corer is developed and one control objective is to maximize the drilling rates via tuning the sinusoidal drive frequency [111]. NOESC is used to tune the driving frequency of a time-varying resonating actuator subjected to both random and high-power impulsive noise disturbance, where hill climbing algorithm is used to rapidly converge to the neighborhood of resonance, and quadratic programming is used for precise resonance tuning.

Via optical phased arrays (OPA), NOESC is used to achieve simultaneously beam steering and wavefront control for free space laser communication system [202]. The control objective is to determine an OPA phase profile that maximizes the power-in-fiber, where quasi-Newton optimization algorithm is used in the NOESC.

Heating, Ventilation and Air Conditioning (HVAC) systems are often designed and specified independently of each other. Therefore, a path to optimize building system energy efficiency is of important value. Focusing on chillers and towers in the building, [203–205] proposed using extremum seeking control to minimize the total power consumption via tuning the condenser water temperature, and perturbation based and numerical optimization based ESC are implemented.

To maximize the user's power output, recent research has focused on closed-loop actuated exercise machines that incorporate the feedback from the user. In [112, 206], a nonlinear exercise machine controller is developed for a single degree of freedom system. A desired trajectory is designed to seek the optimal velocity set-point that will maximize the user's power output, while the controller is designed to ensure that the exercise machine tracks the resulting desired trajectory. This work actually uses the same idea of NOESC presented in this book. In this paper, sinusoidal perturbation and Brents' method or Simplex method are used to generate the tracking trajectory and the tracking controllers are designed based on Lyapunov analysis, where the torque is either measured or estimated. Experimentation results were provided and illustrated the performance of the proposed control strategy. Other studies of exercise machine control via extremum seeking appear in [70, 71].

References

1. Drakunov, S.V., Özgüner, Ü, Dix, P., Ashrafi, B.: ABS control using optimum search via sliding modes. IEEE Transactions on Control Systems Technology (3), 79–85 (1995)
2. Tunay, İ.: Antiskid control for aircraft via extremum-seeking. In: Proceedings of the American Control Conference, vol. 2, pp. 665–670 (2001)
3. Ariyur, K.B., Krstić, M.: Real-time Optimization by Extremum-seeking Control. Wiley-Interscience, Hoboken (2003)
4. Aström, K.J., Wittenmark, B.: Adaptive Control. Wiley, Boston (1994)
5. Wang, H.-H., Krstić, M., Bastin, G.: Optimizing bioreactors by extremum seeking. Int. J. Adapt. Control Signal Process. **13**(8), 651–669 (1999)
6. Zhang, T., Guay, M., Dochain, D.: Adaptive extremum seeking control of continuous stirred-tank bioreactors. AIChE J. **49**(1), 113–123 (2003)
7. Binetti, P., Ariyur, K.B., Krstić, M.: Control of formation flight via extremum seeking. In: Proceedings of the American Control Conference, vol. 4, pp. 2848–2853 (2002)
8. Killingsworth, N., Krstić, M.: PID tuning using extremum seeking. IEEE Control Syst. Mag. **26**, 70–79 (2006)
9. Athans, M., Falb, P.: Optimal Control: an Introduction to the Theory and Its Applications. McGraw-Hill, New York (1966)
10. Kwakernaak, H., Sivan, R.: Linear Optimal Control Systems. Wiley, Boston (1972)
11. Lewis, F., Syrmos, V.: Optimal Control. Wiley, Boston (1995)
12. Allgöwer, F., Zheng, A.: Nonlinear Model Predictive Control. Birkhäuser, Basel (2000)
13. Camacho, E., Bordons, C.: Model Predictive Control. Springer, London (2004)
14. Guay, M., Zhang, T.: Adaptive extremum seeking control of nonlinear dynamic systems with parametric uncertainties. Automatica **39**(7), 1283–1293 (2003)
15. Lavretsky, E., Hovakimyan, N., Calise, A.: Adaptive extremum seeking control design. In: Proceedings of the American Control Conference, vol. 1, pp. 567–572 (2003)
16. Zhou, K., Doyle, J., Glover, K.: Robust and Optimal Control. Prentice Hall, Upper Saddle River (1996)

17. Nusawardhana, Żak, S.H.: Simultaneous perturbation extremum seeking method for dynamic optimization problems. In: Proceedings of the American Control Conference, pp. 2085–2810 (2004)
18. Rotea, M.A.: Analysis of multivariable extremum seeking algorithms. In: Proceedings of the American Control Conference, vol. 1, pp. 433–437 (2000)
19. Banavar, R.N., Chichka, D.F., Speyer, J.L.: Convergence and synthesis issues in extremum seeking control. In: Proceedings of the American Control Conference, vol. 1, pp. 438–443 (2000)
20. Hamza, M.H.: Extremum control of continuous systems. IEEE Trans. Autom. Control **11**(2), 182–189 (1966)
21. Leblanc, M.: Sur l'electrification des chemins de fer au moyen de courants alternatifs de frequence elevee. Rev. Gen. Electr. (1922)
22. Sternby, J.: Extremum control systems: An area for adaptive control? In: Preprints of the Joint American Control Conference, 1980
23. Krstić, M., Wang, H.-H.: Design and stability analysis of extremum seeking feedback for general nonlinear systems. Automatica **36**(2), 595–601 (2000)
24. Krstić, M.: Performance improvement and limitations in extremum seeking control. Syst. Control Lett. **39**(5), 313–326 (2000)
25. Tan, Y., Moase, W.H., Manzie, C., Nesic, D., Mareels, I.M.Y.: Extremum seeking from 1922 to 2010. In: 29th Chinese Control Conference (CCC, 2010), pp. 14–26 (2010)
26. Pan, Y., Özgüner, Ü., Acarman, T.: Stability and performance improvement of extremum seeking control with sliding mode. Int. J. Control **76**(9/10), 968–985 (2003)
27. Nusawardhana, Żak, S.H.: Extremum seeking using analog nonderivative optimizers. In: Proceedings of the American Control Conference, vol. 4, pp. 3242–3247 (2003)
28. Meerkov, S.M.: Asymptotic methods for investigating quasistationary states in continuous systems of automatic optimization. Autom. Remote Control **11**, 1726–1743 (1967)
29. Meerkov, S.M.: Asymptotic methods for investigating a class of forced states in extremal systems. Autom. Remote Control **12**, 1916–1920 (1967)
30. Meerkov, S.M.: Asymptotic methods for investigating stability of continuous systems of automatic optimization subject to disturbance action. Avtom. Telemeh. **12**, 14–24 (1967) (in Russian)
31. Choi, J.-Y., Krstić, M., Ariyur, K.B., Lee, J.S.: Extremum seeking control for discrete-time systems. IEEE Trans. Autom. Control **47**(2), 318–323 (2002)
32. Takata, H., Hachino, T., Tamura, R., Komatsu, K.: Design of extremum seeking control with accelerator. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E88-A**(10), 2535–2540 (2005)
33. Walsh, G.C.: On the application of multi-parameter extremum seeking control. In: Proceedings of the American Control Conference, vol. 1, pp. 411–415 (2000)
34. Ariyur, K.B., Krstić, M.: Analysis and design of multivariable extremum seeking. In: Proceedings of the American Control Conference, vol. 4, pp. 2903–2908 (2002)
35. Ariyur, K.B., Krstić, M.: Sloping seeking: a generalization of extremum seeking. Int. J. Adapt. Control Signal Process. **18**(2), 1–22 (2002)
36. Tan, Y., Nešić, D., Mareels, I.: On non-local stability properties of extremum seeking control. Automatica **42**(6), 889–903 (2006)
37. Tan, Y., Mareels, I., Nešić, D., Xu, J.-X.: Point-wise extremum seeking control under repeatable control environment. In: Proceedings of the 22nd IEEE International Symposium on Intelligent Control, pp. 237–242 (2007)
38. Tan, Y., Nešić, D., Mareels, I.: On the choice of dither in extremum seeking systems: A case study. Automatica **44**(5), 1446–1450 (2008)
39. Tan, Y., Nešić, D., Mareels, I., Astolfi, A.: On global extremum seeking in the presence of local extrema. Automatica **45**(1), 245–251 (2009)
40. Zhang, C., Siranosian, A., Krstić, M.: Extremum seeking for moderately unstable systems and for autonomous target tracking without position measurements. In: Proceedings of the American Control Conference, pp. 4939–4944 (2006)

41. Manzie, C., Krstić, M.: Extremum seeking with stochastic perturbations. IEEE Trans. Autom. Control **54**(3), 580–585 (2009)
42. Liu, S.-J., Krstić, M.: Stochastic averaging in continuous time and its applications to extremum seeking. IEEE Trans. Autom. Control **55**(10), 2235–2250 (2010)
43. Mesquita, A.R., Hespanha, J.P., Aström, K.: Optimotaxis: a stochastic multiagent optimization procedure with point measurements. In: Egerstedt, M., Mishra, B. (eds.) Hybrid Systems: Computation and Control. Lecture Notes in Computer Science, vol. 4981, pp. 358–371 (2008)
44. Moase, W.H., Manzie, C., Brear, M.J.: Newton-like extremum-seeking for the control of thermoacoustic instability. IEEE Trans. Autom. Control **55**(9), 2094–2105 (2010)
45. Korovin, S.K., Utkin, V.I.: Use of the sliding mode in problems of static optimization. Autom. Remote Control **33**, 570–579 (1972)
46. Korovin, S.K., Utkin, V.I.: Using sliding modes in static optimization and nonlinear programming. Automatica **10**, 525–532 (1974)
47. Drakunov, S.V., Özgüner, Ü.: Optimization of nonlinear system output via sliding mode approach. In: Proceedings of the IEEE International Workshop on Variable Structure and Lyapunov Control of Uncertain Dynamical Systems, Sheffield, UK, pp. 61–62 (1992)
48. Winkelman, J., Haskara, I., Özgüner, Ü.: Tuning for dynamic spark advance control. In: Proceedings of the American Control Conference, pp. 163–164 (1999)
49. Haskara, I., Özgüner, Ü., Winkelman, J.: Extremum control for optimal operating point determination and set point optimizing via sliding modes. J. Dyn. Syst. Meas. Control **122**(4), 719–724 (2000)
50. Haskara, I., Hatipoglu, C., Özgüner, Ü.: Sliding mode compensation, estimation and optimization methods in automotive control. In: Yu, X., Xu, J.-X. (eds.) Variable Structure Systems: Towards the 21st Century. Lecture Notes in Control and Information Sciences, vol. 274, 155–174 (2002)
51. Pan, Y., Acarman, T., Özgüner, Ü.: Nash solution by extremum seeking control approach. In: Proceedings of the Conference on Decision and Control, vol. 1, pp. 329–334 (2002)
52. Yu, H., Özgüner, Ü.: Extremum-seeking control via sliding mode with periodic search signals. In: Proceedings of the Conference on Decision and Control, vol. 1, pp. 323–328 (2002)
53. Yu, H., Özgüner, Ü.: Smooth extremum-seeking control via second order sliding mode. In: Proceedings of the American Control Conference, vol. 4, pp. 3248–3253 (2003)
54. Pan, Y., Özgüner, Ü.: Sliding mode extremum seeking control for linear quadratic dynamic game. In: Proceedings of the American Control Conference, pp. 614–619 (2004)
55. Fu, L., Özgüner, Ü.: Variable structure extremum seeking control based on sliding mode gradient estimation for a class of nonlinear systems. In: Proceedings of the American Control Conference, pp. 8–13 (2009)
56. Tan, H.S., Tomizuka, M.: An adaptive sliding mode vehicle traction controller design. In: Proceedings of the American Control Conference, pp. 1035–1058 (1989)
57. Yamanka, S., Ohmori, H.: Nonlinear adaptive extremum seeking control for time delayed index in the presence of deterministic disturbance. In: Proceedings of the PhysCon 2005, pp. 121–125 (2005)
58. Peixoto, A.J., Oliveira, T.R., Hsu, L.: Periodic switching function based sliding mode control applied to output-feedback extremum-seeking problem. In: 11th International Workshop on Variable Structure Systems (VSS, 2010), pp. 124–129 (2010)
59. Chin, Y.K. et al.: Sliding mode ABS wheel slip control. In: Proceedings of the American Control Conference, pp. 1–6 (1992)
60. Haskara, I., Özgüner, Ü., Winkelman, J.: Wheel slip control for antispin acceleration via dynamic spark advance. Control Eng. Pract. **8**(10), 1135–1148 (2000)
61. Yu, H., Özgüner, Ü.: Extremum-seeking control strategy for ABS system with time delay. In: Proceedings of the American Control Conference, vol. 5, pp. 3753–3758 (2002)
62. Gros, S., Srinivasan, B., Bonvin, D.: Optimizing control based on output feedback. Comput. Chem. Eng. **33**, 191–198 (2009)

63. Speyer, J.L., Banavar, R.N., Chichka, D.F., Rhee, I.: Extremum seeking loops with assumed functions. In: Proceedings of the 39th Conference on Decision and Control, vol. 1, pp. 142–147 (2000)
64. Banavar, R.N., Chichka, D.F., Speyer, J.L.: Functional feedback in an extremum seeking loop. In: Proceedings of the 40th Conference on Decision and Control, vol. 2, pp. 1316–1321 (2001)
65. Banavar, R.N.: Extremum seeking loops with assumed functions: estimation and control. In: Proceedings of the American Control Conference, vol. 4, pp. 3159–3164 (2002)
66. Banavar, R.N., Chichka, D.F., Speyer, J.L.: Convergence and synthesis issues in extremum seeking control. Int. J. Adapt. Control Signal Process. **17**, 751–762 (2003)
67. Ryan, J.J., Speyer, J.L.: Peak-seeking control using gradient and hessian estimates. In: Proceedings of the American Control Conference, pp. 611–616 (2010)
68. Spooner, J.T., Maggiore, M., Ordóñez, R., Passino, K.M.: Stable Adaptive Control and Estimation for Nonlinear Systems, Neural and Fuzzy Approximator Techniques. Wiley, New York (2002)
69. Li, P.Y., Horowitz, R.: Self-optimizing control. In: Proceedings of the Conference on Decision and Control, pp. 1228–1233 (1997)
70. Li, P.Y., Horowitz, R.: Control of smart exercise machine—part ii: Self-optimizing control. IEEE/ASME Trans. Mechatron. **2**(4), 248–258 (1997)
71. Shields, J., Horowitz, R.: Controller design and experimental verification of a self-optimizing motion controller for smart exercise machines. In: Proceedings of the American Control Conference, pp. 2736–2742 (1997)
72. Adetola, V., Dehaan, D., Guay, M.: Adaptive extremum seeking receding horizon control of nonlinear system. In: Proceedings of the American Control Conference, pp. 2937–2942 (2004)
73. Adetola, V., Guay, M.: Adaptive output feedback extremum seeking receding horizon control of linear systems. J. Process Control **16**, 521–533 (2006)
74. Dochain, D., Guay, M.: Extremum seeking control of state-constrained nonlinear systems. Automatica **41**, 1567–1574 (2005)
75. Guay, M., Peters, N.: Real-time dynamical optimization of nonlinear systems: A flatness-based approach. Comput. Chem. Eng. **30**, 709–721 (2006)
76. Adetola, V., Guay, M.: Parameter convergence in adaptive extremum-seeking control. Automatica **43**(1), 105–110 (2007)
77. Teixeira, M.C.M., Żak, S.H.: Analog neural nonderivative optimizers. IEEE Trans. Neural Netw. **9**(4), 629–638 (1998)
78. Nesic, D., Mohammadi, A., Manzie, C.: A systematic approach to extremum seeking based on parameter estimation. In: Proceedings of the Conference on Decision and Control, pp. 3902–3907 (2010)
79. Nesic, D., Tan, Y., Moase, W.H., Manzie, C.: A unifying approach to extremum seeking: Adaptive schemes based on estimation of derivatives. In: Proceedings of the Conference on Decision and Control, pp. 4625–4630 (2010)
80. Zhang, Y.: Stability and performance tradeoff with discrete time triangular search minimum seeking. In: Proceedings of the American Control Conference, vol. 1, pp. 423–427 (2000)
81. Teel, A.R., Popović, D.: Solving smooth and nonsmooth multivariable extremum seeking problems by the methods of nonlinear programming. In: Proceedings of the American Control Conference, vol. 3, pp. 2394–2399 (2001)
82. Zhang, C., Ordóñez, R.: Numerical optimization-based extremum seeking control of LTI systems. In: Proceedings of the Conference on Decision and Control, pp. 4428–4433 (2005)
83. Zhang, C., Ordóñez, R.: Non-gradient extremum seeking control of feedback linearizable systems with application to ABS design. In: Proceedings of the Conference Decision and Control, pp. 6666–6671 (2006)
84. Zhang, C., Ordóñez, R.: Numerical optimization-based extremum seeking control with application to ABS design. IEEE Trans. Autom. Control **52**(3), 454–467 (2007)
85. Zhang, C., Ordóñez, R.: Robust and adaptive design of numerical optimization-based extremum seeking control. Automatica **45**, 634–646 (2009)

86. Yu, S.-J., Chen, H., Kong, L.: Particle swarm optimization-based extremum seeking control. In: Advanced Intelligent Computing Theories and Applications. Lecture Notes in Computer Science, vol. 6215, pp. 185–196 (2008)
87. Liu, H., Chen, H., Kong, L.: Bacteria foraging optimization-based extremum seeking control. In: IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS, 2010), vol. 3, pp. 110–115 (2010)
88. Li, Y., Shen, J.: Extremum seeking predictive control of LTI systems based on numerical optimization. In: 8th IEEE International Conference on Control and Automation (ICCA, 2010), pp. 1321–1325 (2010)
89. Kapadia, A., Nath, N., Burg, T.C., Dawson, D.M.: Lyapunov-based continuous-stirred tank bioreactor control to maximize biomass production using the Haldane and Monod specific growth models. In: Proceedings of the American Control Conference, pp. 6734–6739 (2010)
90. Ivakhnenko, A.G.: On constructing an extremum controller without hunting oscillations. IEEE Trans. Autom. Control **12**(2), 144–153 (1967)
91. Wellstead, P.E., Scotson, P.G.: Self-tuning extremum control. IEE Proc. Part D. Control Theory Appl. **137**(3), 165–175 (1990)
92. Wellstead, P.E., Scotson, P.G.: Self-tuning optimization of spark ignition automotive engines. IEEE Control Syst. Mag. **10**(3), 94–101 (1990)
93. Floretin, J.J.: An approximately optimal extremal regulator. J. Electron. Control **17**(2), 211–310 (1964)
94. Nascimento, C.L., Zarrop, M.B., Muir, A.: A neural network extremum controller. In: Proceedings of the European Control Conference, Groningen, The Netherlands (1993)
95. Zarrop, M.B., Rommens, M.J.J.: Convergence of a multi-input adaptive extremum controller. In: IEE Proceedings-D, vol. 140 (1993)
96. Bamberger, W., Isermann, R.: Adaptive on-line steady-state optimization of slow dynamic processes. Automatica **14**, 223–232 (1978)
97. Navarro, L.B., Zarrop, M.B.: Dynamic extremum control. In: Proceedings of the IEE Colloquium on Adaptive Control, London (1996)
98. Navarro, L.B., Zarrop, M.B.: Adaptive implicit control using extremum control methods. In: Proceedings of the UKACC International Conference on Control (1996)
99. Wittenmark, B., Evans, R.J.: Extremal control of wiener model processes. In: Proceedings of the Conference on Decision and Control, pp. 4637–4642 (2002)
100. Velut, S., Hagander, P.: Analysis of probing control strategy. In: Proceedings of the American Control Conference, vol. 4, pp. 609–614 (2003)
101. Velut, S., Hagander, P.: A probing control strategy: stability and performance. In: Proceedings of the Conference on Decision and Control, pp. 4607–4612 (2004)
102. Pan, Y., Özgüner, Ü.: Discrete-time extremum seeking algorithms. In: Proceedings of the American Control Conference, vol. 4, pp. 3147–3152 (2002)
103. Egardt, B., Larsson, S.: On a parameter adaptive extremum controller. In: Proceedings of the Conference on Decision and Control, pp. 4809–4814 (2005)
104. Banaszuk, A., Ariyur, K.B., Krstić, M., Jacobson, C.A.: An adaptive algorithm for control of combustion instability and application to compressor instability control. Automatica **40**(11), 1965–1972 (2004)
105. Li, Y., Rotea, M.A., Chiu, G.T.-C., Mongeau, L.G., Paek, I.-S.: Extremum seeking control of a tunable thermoacoustic cooler. IEEE Trans. Control Syst. Technol. **13**, 527–536 (2005)
106. Centioli, C., Iannone, F., Panella, M., Pangione, L., Podda, S., Tuccillo, A., Zaccarian, L.: Advances in the experimentation of extremum seeking techniques to maximize the RF power absorption in FTU. In: Symposium on Fusion Engineering (SOFE) (2005)
107. Popović, D., Janković, M., Magner, S., Teel, A.: Extremum seeking methods for optimization of variable cam timing engine operation. IEEE Trans. Control Syst. Technol. **14**(3), 398–407 (2006)
108. Centioli, C., Iannone, F., Mazza, G., Panella, M., Pangione, L., Podda, S., Tuccillo, A.: Maximization of the lower hybrid power coupling in the Frascati Tokamak upgrade via extremum seeking. Control Eng. Pract. **16**, 1468–1478 (2008)

109. Nakadoi, H., Sobey, D., Yamakita, M., Mukai, T.: Liquid environment-adaptive IPMC fish-like robot using extremum seeking feedback. In: Proceedings of 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3089–3094 (2008)
110. Killingsworth, N., Aceves, S., Flowers, D., Espinosa-Loza, F., Krstić, M.: HCCI engine combustion-timing control: Optimizing gains and fuel consumption via extremum seeking. IEEE Transcations on Control Systems Technology **17**(6) (2009)
111. Aldrich, J., Sherrit, S., Bao, X., Bar-Cohen, Y., Badescu, M., Chang, Z.: Extremum-seeking control for an ultrasonic/sonic driller/corer (USDC) driven at high power. In: Proceedings of SPIE Smart Structures and Materials 2006: Modeling, Signal Processing, and Control, vol. 6166, pp. 616–618 (2006)
112. Zhang, X.T., Dawson, D.M., Dixon, W.E., Xian, B.: Extremum-seeking nonlinear controllers for a human exercise machine. IEEE/ASME Trans. Mechatron. **11**(2), 233–240 (2006)
113. Brunton, S.L., Rowley, C.W., Kulkarni, S.R., Clarkson, C.: Maximum power point tracking for photovoltaic optimization using ripple-based extremum seeking control. IEEE Trans. Power Electron. **25**(10), 2531–2540 (2010)
114. Sato, T., Araki, N., Konishi, Y., Ishigaki, H.: Discrete-time weigh feeder control using extremum-seeking method. In: IEEE International Conference on Control Applications (CCA, 2010), pp. 53–58 (Sept. 2010)
115. van der Meulen, S., de Jager, B., Veldpaus, F., Steinbuch, M.: Combining extremum seeking control and tracking control for high-performance CVT operation. In: Proceedings of the Conference on Decision and Control, pp. 3668–3673 (2010)
116. Zhang, C., Siranosian, A., Krstić, M.: Extremum seeking for moderately unstable systems and for autonomous target tracking without position measurements. Automatica **43**, 1832–1839 (2007)
117. Zhang, C., Arnold, D., Ghods, N., Siranosian, A., Krstić, M.: Source seeking with nonholonomic unicycle without position measurement and with tuning of forward velocity. Syst. Control Lett. **56**, 245–252 (2007)
118. Cochran, J., Krstić, M.: Nonholonomic source seeking with tuning of angular velocity. IEEE Trans. Autom. Control **54**(4), 713–731 (2009)
119. Ghods, N., Krstić, M.: Speed regulation in steering-based source seeking. Automatica **46**, 452–459 (2010)
120. Ghods, N., Krstić, M.: Source seeking with very slow or drifting sensors. ASME J. Dyn. Syst. Meas. Control **133** (2011)
121. Stankovic, M.S., Stipanovic, D.M.: Stochastic extremum seeking with applications to mobile sensor networks. In: Proceedings of the American Control Conference, pp. 5622–5627 (2009)
122. Stankovic, M.S., Stipanovic, D.M.: Discrete time extremum seeking by autonomous vehicles in a stochastic environment. In: Proceedings of the Conference on Decision and Control, pp. 4541–4546 (2009)
123. Liu, S.-J., Krstić, M.: Stochastic source seeking for nonholonomic unicycle. IEEE Trans. Autom. Control **46**, 1443–1453 (2010)
124. Cochran, J., Krstić, M.: 3-d source seeking for underactuated vehicles without position measurement. IEEE Trans. Robot. **25**(1), 117–129 (2009)
125. Cochran, J., Kanso, E., Krstic, M.: Source seeking for a three-link model of fish locomotion. In: Proceedings of the American Control Conference, pp. 1808–1813 (2009)
126. Cochran, J., Kanso, E., Kelly, S.D., Xiong, H., Krstić, M.: Source seeking for two nonholonomic models of fish locomotion. IEEE Trans. Robot. **25**(5), 1166–1176 (2009)
127. Krstić, M., Cochran, J.: Extremum seeking for motion optimization: from bacteria to nonholonomic vehicles. In: Proceedings of the Chinese Control and Decision Conference, pp. 18–27 (2008)
128. Fu, L., Özgüner, Ü.: Sliding mode in constrained source tracking with non-holonomic vehicles. In: International Workshop on Variable Structure Systems (VSS'08), pp. 30–34 (2008)
129. Fu, L., Özgüner, Ü.: Extremum-seeking control in constrained source tracing with nonholonomic vehicles. IEEE Trans. Ind. Electron. **56**(9), 3602–3608 (2009)

130. Mayhew, C.G., Sanfelice, R.G., Teel, A.R.: Robust source-seeking hybrid controller for autonomous vehicles. In: Proceedings of the American Control Conference, pp. 1185–1190 (2007)
131. Biyik, E., Arcak, M.: Gradient climbing in formation via extremum seeking and passivity-based coordination rules. In: Proceedings of the Conference on Decision and Control, pp. 3133–3138 (2007)
132. Ghods, N., Frihauf, P., Krstic, M.: Multi-agent deployment in the plane using stochastic extremum seeking. In: Proceedings of the Conference on Decision and Control (2010)
133. Ghods, N., Krstic, M.: Multiagent deployment over a source. IEEE Trans. Control Syst. Technol., pp. 1–10 (2011)
134. Dixon, C., Frew, E.W.: Controlling the mobility of network nodes using decentralized extremum seeking. In: Proceedings of the Conference on Decision and Control, pp. 1291–1296 (2006)
135. Chichka, D.F., Speyer, J.L., Park, C.G.: Peaking-seeking control with application to formation flight. In: Proceedings of the 38th Conference on Decision and Control, vol. 5, pp. 2463–2470 (1999)
136. Xie, F., Zhang, X., Fierro, R., Motter, M.: Autopilot-based nonlinear UAV formation controller with extremum-seeking. In: Proceedings of the Conference on Decision and Control, pp. 4933–4938 (2005)
137. Popović, D., Janković, M., Magner, S., Teel, A.: Extremum seeking methods for optimization of variable cam timing engine operation. In: Proceedings of the American Control Conference, vol. 4, pp. 3136–3141 (2003)
138. Sugihira, S., Ichikawa, K., Ohmori, H.: Starting speed control of SI engine based on online extremum control. In: SICE Annual Conference 2007, pp. 2569–2573 (2007)
139. Haskara, Í., Zhu, G.G., Winkelman, J.: Multivariable EGR/spark timing control for IC engines via extremum seeking. In: Proceedings of the American Control Conference, pp. 163–164 (1999)
140. Gafvert, M., Arzen, K.-E., Pederson, L.M.: Simple linear feedback and extremum control of GDI engines. In: Seoul 2000 FISITA World Automotive Congress (2000)
141. Janković, M., Magner, S.: Optimization and scheduling for automotive powertrains. In: Proceedings of the American Control Conference, pp. 4054–4059 (2004)
142. Tanelli, M., Astolfi, A., Savaresi, S.M.: Non-local extremum seeing control for active braking control systems. In: Proceedings of the American Control Conference, pp. 891–896 (2006)
143. Wang, H.-H., Yeung, S., Krstić, M.: Experimental application of extremum seeking on an axial-flow compressor. IEEE Trans. Control Syst. Technol. **8**(2), 300–309 (2000)
144. Schneider, G., Ariyur, K.B., Krstić, M.: Tuning of a combustion controller by extremum seeking: a simulation study. In: Proceedings of the 39th Conference on Decision and Control, vol. 5, pp. 5219–5223 (2000)
145. Banaszuk, A., Zhang, Y., Jacobson, C.A.: Adaptive control of combustion instability using extremum-seeking. In: Proceedings of the American Control Conference, vol. 1, pp. 416–422 (2000)
146. Schuster, E., Romero, C., Yao, Z., Si, F.: Integrated real-time optimization of boiler and post-combustion system in coal-based power plants via extremum seeking. In: IEEE International Conference on Control Applications (CCA, 2010), pp. 2184–2189 (2010)
147. Marcos, N.I., Guay, M., Dochain, D.: Output feedback adaptive extremum seeking control of a continuous stirred tank bioreactor with Monod's kinetics. J. Process Control **14**(11), 807–818 (2004)
148. Marcos, N.I., Guay, M., Dochain, D., Zhang, T.: Adaptive extremum-seeking control of a continuous stirred tank bioreactor with Haldane's kinetics. J. Process Control **14**(3), 317–328 (2004)
149. Titica, M., Dochain, D., Guay, M.: Adaptive extremum-seeking control of fed-batch bioreactors. Eur. J. Control **9**(6), 614–627 (2003)
150. Peters, N., Guay, M., DeHaan, D.: Real-time dynamic optimization of batch systems. J. Process Control **17**(3), 261–271 (2007)

151. Guay, M., Dochain, D., Perrier, M.: Adaptive extremum seeking control of continuous stirred tank bioreactors with unknown growth kinetics. Automatica **40**(5), 881–888 (2004)
152. Guay, M., Dochain, D., Perrier, M.: Adaptive extremum-seeking control of nonisothermal continuous stirred tank reactors. Chem. Eng. Sci. **60**(13), 3671–3681 (2005)
153. Hudon, N., Perrier, M., Guay, M., Dochain, D.: Adaptive extremum seeking control of a non-isothermal tubular reactor with unknown kinetics. Comput. Chem. Eng. **29**, 839–849 (2005)
154. Cougnon, P., Dochain, D., Guay, M., Perrier, M.: Real-time optimization of a tubular reactor with distributed feed. AIChE J. **52**(6), 2120–2128 (2006)
155. Hudon, N., Guay, M., Perrier, M., Dochain, D.: Adaptive extremum seeking control of tubular reactor with limited actuation. In: Proceedings of the American Control Conference, pp. 4563–4568 (2005)
156. Hudon, N., Guay, M., Perrier, M., Dochain, D.: Adaptive extremum-seeking control of convection-reaction distributed reactor with limited actuation. Comput. Chem. Eng. (Jan. 2008)
157. Bastin, G., Nešić, D., Tan, Y., Mareels, I.: On extremum seeking in bioprocesses with multivalued cost functions. Biotechnol. Prog. **25**(3), 683–689 (2009)
158. Nguang, S.K., Cheng, X.D.: Extremum seeking scheme for continuous fermentation processes described by an unstructured fermentation model. Bioprocess Biosyst. Eng. **23**(5), 417–420 (2000)
159. Beaudoin, J.F., Cadot, O., Aider, J.L., Wesfreid, J.E.: Bluff-body drag reduction by extremum-seeking control. J. Fluids Struct. **22**(6–7), 973–978 (2006)
160. Creaby, J., Li, Y., Seem, J.E.: Maximizing wind turbine energy capture using multivariable extremum seeking control. Wind Eng. **33**(4), 361–387 (2009)
161. Kim, K., Kasnakoglu, C., Serrani, A., Samimy, M.: Extremum-seeking control of subsonic cavity flow. AIAA J. **47**(1), 195–205 (2009)
162. Sinha, A., Kim, K., Kim, J.-H., Serrani, A., Samimy, M.: Extremizing feedback control of a high-speed and high Reynolds number jet. AIAA J. **48**(2), 387–399 (2010)
163. Zhang, T., Wen, J.T., Julius, A., Peles, Y., Jensen, M.K.: Extremum seeking micro-thermal-fluid control for active two-phase microelectronics cooling. In: Proceedings of the Conference on Decision and Control, pp. 1899–1904 (2010)
164. Luo, L., Schuster, E.: Boundary feedback control for heat exchange enhancement in 2d magnetohydrodynamic channel flow by extremum seeking. In: Proceedings of the Conference on Decision and Control, pp. 8272–8277 (2009)
165. Luo, L., Schuster, E.: Heat exchange enhancement by extremum seeking boundary feedback control in 3d magnetohydrodynamic channel flow. In: Proceedings of the Conference on Decision and Control, pp. 2972–2978 (2010)
166. Banaszuk, A., Narayanan, S., Zhang, Y.: Adaptive control of flow separation in a planar diffuser. In: 41st Aerospace Sciences Meeting & Exhibit (2003)
167. Peterson, K.S., Stefanopoulou, A.G.: Extremum seeking control for soft landing of an electromechanical valve actuator. Automatica **40**(6), 1063–1069 (2004)
168. Centioli, C., Iannone, F., Mazza, G., Panella, M., Pangione, L., Podda, S., Tuccillo, A., Vitale, V., Zaccarian, L.: Extremum seeking applied to the plasma control system of the Frascati tokamak upgrade. In: Proceedings of the Conference on Decision and Control, pp. 8227–8232 (2005)
169. Carnevale, D., Zaccarian, L., Astolfi, A., Podda, S.: Extremum seeking without external dithering and its application to plasma RF heating on FTU. In: Proceedings of the Conference on Decision and Control, pp. 3151–3156 (2008)
170. Ou, Y., Xu, C., Schuster, E., Luce, T., Ferron, J.R., Walker, M.: Extremum-seeking finite-time optimal control of plasma current profile at the diii-d tokamak. In: Proceedings of the American Control Conference, pp. 4015–4020 (2007)
171. Cardenas, R., Pena, R.: Sensorless vector control of induction machines for variable-speed wind energy applications. IEEE Trans. Energy Convers. **19**(1), 196–205 (2004)

172. Liu, Q., He, Y., Zhao, R.: The maximal wind-energy tracking control of a variable-speed constant-frequency wind-power generation system. Autom. Electr. Power Syst. **27**(20), 62–67 (2003)
173. Geng, H., Yang, G.: A novel control strategy of MPPT taking dynamics of wind turbine into account. In: Proceedings of the 37th IEEE Power Electronic Specialists Conference, pp. 1–6 (2006)
174. Leyva, R., Alonso, C., Queinnec, I., Cid-Pastor, A., Lagrange, D., Martínez-Salamero, L.: MPPT of photovoltaic systems using extremum-seeking control. IEEE Trans. Aerosp. Electron. Syst. **42**(1), 249–258 (2006)
175. Pan, T., Ji, Z., Jiang, Z.: Maximum power point tracking of wind energy conversion systems based on sliding mode extremum seeking control. In: Proceedings of the IEEE Energy Conference, pp. 1–5 (2008)
176. Zheng, X., Li, L., Xu, D., Platts, J.: Sliding mode MPPT control of variable speed wind power system. In: Asia-Pacific Power and Energy Engineering Conference (APPEEC 2009), pp. 1–4 (2009)
177. Bizon, N.: On tracking robustness in adaptive extremum seeking control of the fuel cell power plants. Appl. Energy **87**(10), 3115–3130 (2010)
178. Moura, S.J., Chang, Y.A.: Asymptotic convergence through Lyapunov-based switching in extremum seeking with application to photovoltaic systems. In: Proceedings of the American Control Conference, pp. 3542–3548 (2010)
179. Fu, D., Xing, Y., Ma, Y.: MPPT of VSCF wind energy conversion system using extremum control strategy. In: World Non-Grid-Connected Wind Power and Energy Conference (WNWEC, 2010), pp. 1–6 (2010)
180. Lu, J., Zahedi, A.: Maximum efficiency point tracking control for fuel cell power systems. In: International Conference on Power System Technology (POWERCON, 2010), pp. 1–6 (2010)
181. Lei, P., Li, Y., Seem, J.: Sequential ESC based global MPPT control for photovoltaic array with variable shading. IEEE Trans. Sustain. Energy (2011)
182. Bratcu, A.I., Munteanu, I., Bacha, S., Picault, D., Raison, B.: Cascaded dc-dc converter photovoltaic systems: Power optimization issues. IEEE Trans. Ind. Electron. **58**(2), 403–411 (2011)
183. Chang, Y.A., Moura, S.J.: Air flow control in fuel cell systems: An extremum seeking approach. In: Proceedings of the American Control Conference, pp. 1052–1059 (2009)
184. Komatsu, H.O.M., Miyamoto, H., Sano, A.: Output maximization control of wind turbine based on extremum control strategy. In: Proceedings of the American Control Conference, pp. 1739–1740 (2001)
185. Li, Y., Rotea, M.A., Chiu, G.T.-C., Mongeau, L.G., Paek, I.-S.: Extremum seeking control of tunable thermoacoustic cooler. In: Proceedings of the American Control Conference, pp. 2033–2038 (2004)
186. Dower, P., Farrell, P., Nešić, D.: Extremum seeking control of cascaded Raman optical amplifiers. IEEE Trans. Control Syst. Technol. **16**(3), 396–407 (2008)
187. Schuster, E., Morinaga, E., Allen, C.K., Krstić, M.: Optimal beam matching in particle accelerators via extremum seeking. In: Proceedings of the American Control Conference, pp. 1962–1967 (2006)
188. Schuster, E., Xu, C., Torres, N.: Extremum seeking adaptive control of beam envelope in particle accelerators. In: Proceedings of the IEEE Conference on Control Applications (2006)
189. Schuster, E., Xu, C., Torres, N., Morinaga, E., Allen, C., Krstić, M.: Beam matching adaptive control via extremum seeking. Nucl. Instrum. Methods Phys. Res., Sect. A, Accel. Spectrom. Detect. Assoc. Equip. **581**, 799–815 (2007)
190. Cistelecan, M.R.: Power control in mobile wireless network using sliding mode extremum seeking control implementing bifurcations. In: Proceedings of 17th IEEE International Conference on Control Applications, pp. 67–72 (2008)
191. Pan, Y., Furuta, K.: Variable structure control with sliding sector based on hybrid switching law. Int. J. Adapt. Control Signal Process. **21**(8–9), 764–778 (2007)

192. Schuster, E., Sondak, D., Arastoo, R., Walker, M.L., Humphreys, D.A.: Optimal tuning of tokamak plasma equilibrium controllers in the presence of time delays. In: IEEE International Conference on Control Applications (CCA), Intelligent Control (ISIC, 2009), pp. 1188–1194 (2009)
193. Zuo, B., Hu, Y.-A., Li, J.: Pid controller tuning by using extremum seeking algorithm based on annealing recurrent neural network. In: 3rd International Symposium on Knowledge Acquisition and Modeling (KAM, 2010), pp. 132–135 (2010)
194. Dong, W., Hu, Y.-A., Zuo, B., Dong, W.: Backstepping control optimized by the multi-parameter ESA. In: International Conference on Artificial Intelligence and Education (ICAIE, 2010), pp. 155–158 (Oct. 2010)
195. Frihauf, P., Krstic, M., Basar, T.: Nash equilibrium seeking for games with non-quadratic payoffs. In: Proceedings of the Conference on Decision and Control, pp. 881–886 (2010)
196. Krstic, M., Frihauf, P., Krieger, J., Basar, T.: Nash equilibrium seeking with finitely- and infinitely-many players. In: Proceedings of the 8th IFAC Symposium on Nonlinear Control Systems (2010)
197. Stankovic, M.S., Johansson, K.H., Stipanovic, D.M.: Distributed seeking of Nash equilibria in mobile sensor networks. In: Proceedings of the Conference on Decision and Control, pp. 5598–5603 (2010)
198. Wang, H.-H., Krstić, M.: Extremum seeking for limit cycle minimization. IEEE Trans. Autom. Control **45**(12), 2432–2436 (2000)
199. Johansen, T.A., Sbárbaro, D.: Lyapunov-based optimizing control of nonlinear blending processes. IEEE Trans. Control Syst. Technol. **13**(4), 631–638 (2005)
200. Antonello, R., Oboe, R.: Stability analysis of an extremum seeking controller for mode-matching in vibrating microgyros. In: The 8th IEEE International Workshop on Advanced Motion Control, pp. 116–121 (2008)
201. Antonello, R., Oboe, R., Prandi, L., Biganzoli, F.: Automatic mode matching in MEMS vibrating gyroscopes using extremum-seeking control. IEEE Trans. Ind. Electron. **56**(10), 3880–3891 (2009)
202. Danielson, C., Lacy, S., Hindman, B., Collier, P., Hunt, J., Moser, R.: Extremum seeking control for simultaneous beam steering and wavefront correction. In: Proceedings of the American Control Conference, pp. 4483–4488 (2006)
203. Sane, H.S., Haugstetter, C., Bortoff, S.A.: Building HVAC control systems—role of controls and optimization. In: Proceedings of the American Control Conference, pp. 1121–1126 (2006)
204. Li, P., Li, Y., Seem, J.E.: Extremum seeking control for efficient and reliable operation of airside economizers. In: Proceedings of the American Control Conference, pp. 20–25 (2009)
205. Tyagi, V., Sane, H.S., Darbha, S.: An extremum seeking algorithm for determining the set point temperature for condensed water in a cooling tower. In: Proceedings of the American Control Conference, pp. 1127–1131 (2006)
206. Zhang, X.T., Dawson, D.M., Dixon, W.E., Xian, B.: Extremum seeking nonlinear controllers for a human exercise machine. In: Proceedings of the 43th Conference on Decision and Control, pp. 3950–3955 (2004)

Chapter 2
Numerical Optimization

2.1 Mathematical Background

In this chapter, we present the necessary mathematical background on numerical optimization [11], which is used in this book as a fundamental tool for extremum seeking control. We first review concepts related to continuity, differentiability, and optimality. These concepts will then allow us to present the line search and trust-region unconstrained optimization methods.

Definition 2.1.1 (Sequence) A sequence of real numbers $\{x_k^s | k = 1, 2, \dots\}$ (also represented as $\{x_k^s\}$) is said to *converge* to a limit $x \in \mathbb{R}$ if for every $\varepsilon > 0$ there exists some positive integer K (that depends on ε) such that, for every $k \geq K$, we have $|x_k^s - x| < \varepsilon$. For such a convergent sequence we may also write $\lim_{k\to\infty} x_k^s = x$.

Similarly, a sequence $\{x_k^s\}$ of vectors $x_k^s \in \mathbb{R}^n$ is said to converge to a limit $x \in \mathbb{R}^n$ if the ith coordinate of x_k^s converges to the ith coordinate of x for $1 \leq i \leq n$. In this case, the notation $\lim_{k\to\infty} x_k^s = x$ is employed as well.

Definition 2.1.2 (Continuously Differentiable Functions) Consider the function $f : \mathbb{R} \to \mathbb{R}$. This function is said to be *continuously differentiable* if its derivative f' exists and is continuous. Alternatively, one can say that f *belongs to class* C^1, or $f \in C^1$.

Similarly, if f', f'', $\dots$, $f^{(k)}$ exist and are continuous, then f is said to belong to *class* C^k, or $f \in C^k$. Finally, if f has continuous derivatives of all orders, then it is said to be *smooth*, or $f \in C^\infty$.

For a multivariate function $f : \mathbb{R}^n \to \mathbb{R}$, it is said to belong to class C^k if each of its derivatives up to kth order exists and is continuous.

In addition to these continuity concepts, there is another type of continuity we often need, called Lipschitz continuity. Lipschitz continuity is a smoothness condition, stronger than regular continuity, that imposes a limit on the function's growth rate.

C. Zhang, R. Ordóñez, *Extremum-Seeking Control and Applications*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2224-1_2, © Springer-Verlag London Limited 2012

Definition 2.1.3 (Lipschitz Continuity) Consider the function $f : \mathbb{R}^n \to \mathbb{R}^m$. This function is said to be *Lipschitz continuous* if there exists a constant $L > 0$ such that

$$\|f(x) - f(y)\| \leq L\|x - y\|. \tag{2.1}$$

The following cases can be distinguished:

1. If condition (2.1) is satisfied for every $x, y \in \mathbb{R}^n$ and with *the same* constant L, then f is said to be *globally Lipschitz continuous*, or simply *Lipschitz*.
2. If condition (2.1) is satisfied for every $x, y \in D$, where $D \in \mathbb{R}^n$ and with the same constant L, then f is said to be *Lipschitz in D*.
3. If condition (2.1) is satisfied for every $x, y \in D$, where $D \in \mathbb{R}^n$ but not with the same constant L, which can depend on x, then f is said to be *locally Lipschitz in D*.

For the study of nonlinear systems it is often useful to define functions that map one set to another, with the purpose of normalizing the form of a system's dynamics. To this end, we require the following concept.

Definition 2.1.4 (Diffeomorphism) Consider a function $T : D \to M$ mapping the domain $D \in \mathbb{R}^n$ to the domain $M \in \mathbb{R}^n$. The function T is called a *diffeomorphism* if T is continuously differentiable, and its inverse T^{-1} exists and is continuously differentiable.

Definition 2.1.5 (Matrix Concepts) An $n \times n$ matrix A with real elements is *symmetric* if it is equal to its transpose, or $A = A^\top$. The symmetric matrix A is said to be *positive definite* if $x^\top Ax > 0$ for all $x \in \mathbb{R}^n$, $x \neq 0$. A is called *positive semi-definite* if $x^\top Ax \geq 0$. A is *negative definite* if $-A$ is positive definite.

Theorem 2.1.6 *The following statements hold for any non-zero symmetric matrix $A \in \mathbb{R}^{n\times n}$* [8]:

1. *Its eigenvalues are all real-valued, and the corresponding n eigenvectors are real, non-zero and mutually orthogonal.*
2. *A is positive definite* if and only if *all its eigenvalues are strictly positive.*
3. *A is positive semi-definite* if and only if *all its eigenvalues are non-negative.*

Given a function $J : \mathbb{R}^n \to \mathbb{R}$ with $J \in C^1$, denote

$$g(x) = \nabla J(x) = \left[\frac{\partial J(x)}{\partial x_1}, \dots, \frac{\partial J(x)}{\partial x_n}\right]^\top$$

as the *gradient* of J evaluated at the point $x \in \mathbb{R}^n$. In the case of $x = x(t)$ for $t \in \mathbb{R}^m$, by using the chain rule we have

$$\nabla J\big(x(t)\big) = \sum_{i=1}^{n} \frac{\partial J}{\partial x_i} \nabla x_i(t) = \sum_{i=1}^{n} \frac{\partial J}{\partial x_i}\left[\frac{\partial x_i(t)}{\partial t_1}, \dots, \frac{\partial x_i(t)}{\partial t_m}\right]^\top.$$

If $J \in C^2$, let $H(x) = \nabla^2 J(x)$ be the *Hessian matrix*. The (i, j)th component of H is given by $\partial^2 J(x)/\partial x_i \partial x_j$, $1 \le i, j \le n$. The square matrix H is symmetric.

Theorem 2.1.7 (Taylor's Theorem [11]) *Let $J : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and $p \in \mathbb{R}^n$. Then*

$$J(x + p) = J(x) + \nabla J^\top (x + \alpha p)p,$$

for some $\alpha \in [0, 1]$. Moreover, if J is twice continuously differentiable then

$$\nabla J(x + p) = \nabla J(x) + \left(\int_0^1 \nabla^2 J(x + \tau p)\, d\tau \right) p,$$

and the following three statements hold:

1. *There exists some $\alpha \in [0, 1]$ such that, for any $p \in \mathbb{R}^n$,*

$$J(x + p) = J(x) + \nabla J(x)^\top p + \frac{1}{2} p^\top \nabla^2 J(x + \alpha p)p.$$

2. *For any $p \in \mathbb{R}^n$,*

$$J(x + p) = J(x) + \nabla J(x)^\top p + \frac{1}{2} p^\top \nabla^2 J(x)p + O\big(\|p\|^2\big).$$

3. *For any $p \in \mathbb{R}^n$,*

$$J(x + p) = J(x) + \nabla J(x)^\top p + \frac{1}{2} p^\top \left(\int_0^1 \left(\int_0^t \nabla^2 J(x + \tau y)\, d\tau \right) dt \right) p.$$

Note that a set $S \subseteq \mathbb{R}^n$ is *convex* if for any $x, y \in S$ we have $\beta x + (1 - \beta)y \in S$ for all $\beta \in [0, 1]$. The set S is *closed* if it contains all of its limit points, and it is *bounded* if all its elements have coordinates whose magnitude is less than some finite $d > 0$. Further, S is *compact* if every sequence of elements of S has a subsequence that converges to an element of S. The set S is compact if and only if it is closed and bounded.

A function J is a *convex function* if its domain is convex and if for any x, y in this domain we have $J(\beta x + (1 - \beta)y) \le \beta J(x) + (1 - \beta)J(y)$ for all $\beta \in [0, 1]$.

Definition 2.1.8 (Global Minimizer) Let J be defined on $S \subseteq \mathbb{R}^n$. The point $x^* \in S$ is a *global minimizer* of J if $J(x^*) \le J(x)$ for all $x \in S$; it is a *strict global minimizer* of J if $J(x^*) < J(x)$ for all $x \in S$, $x \ne x^*$. Correspondingly, we say that $J(x^*)$ is a (*strict*) *global minimum* of J.

Definition 2.1.9 (Local Minimizer) Let J be defined on $S \subseteq \mathbb{R}^n$. The point $x^* \in S$ is a *local minimizer* of J if there exists an open neighborhood B of x^* such that $J(x^*) \le J(x)$ for all $x \in B \cap S$; it is a *strict local minimizer* if $J(x^*) < J(x)$ for all $x \in B \cap S$, $x \ne x^*$. Correspondingly, we say that $J(x^*)$ is a (*strict*) *local minimum* of J.

Definition 2.1.10 (Stationary Point) We say that x^* is a *stationary point* of J defined on $S \subseteq \mathbb{R}^n$ if $\nabla J(x^*) = 0$.

Definition 2.1.11 (Level Set) Let J be defined on $\mathbb{R}^n$ and $\gamma > 0$. We define the *level set with respect to* γ as

$$\mathcal{L}_\gamma = \{x \in \mathbb{R}^n : J(x) \le \gamma\}.$$

Computational procedures that minimize (maximize) J, that is, search for its minima (maxima), are referred to as *optimization methods*. The optimization is often achieved via different *iterative algorithms*. The algorithms begin with a *initial guess* x_0^s and generate a *sequence* $\{x_k^s\}$ leading to a possible solution, that is, a stationary point, a local minimizer or a global minimizer.

Definition 2.1.12 (Algorithm Convergence) Let $S \subseteq \mathbb{R}^n$ and $\{x_k^s\} \subseteq S$ be a sequence generated by an optimization algorithm. If $\lim_{k\to\infty} x_k^s = x^* \in S$ for any $x_0^s \in S$, then we say that the algorithm is *globally convergent*. If such a convergence only exists for some $x_0^s \in S$, then we say the algorithm is *locally convergent*.

Definition 2.1.13 (q-order Convergence) Let $\{x_k^s\}$ be a locally convergent sequence in $S \subseteq \mathbb{R}^n$. We say that $\{x_k^s\}$ is *q-order convergent* if

$$\lim_{k\to\infty} \frac{\|x_{k+1}^s - x^*\|}{\|x_k^s - x^*\|^q} = M$$

exists for some $q, M > 0$. In particular, we say that $\{x_k^s\}$ is *linearly convergent* if $q = 1$; and *superlinearly* or *quadratically convergent* if $1 < q < 2$ or $q = 2$, respectively.

The following standard *stopping criteria* are frequently employed in optimization computations. In the case when $x_k^s \ne 0$ and $J(x_k^s) \ne 0$ for sufficiently large k, computation terminates when

$$\|x_{k+1}^s - x_k^s\| / \|x_k^s\| \le \varepsilon,$$

or

$$|J(x_k^s) - J(x_{k+1}^s)| / |J(x_k^s)| \le \varepsilon.$$

Otherwise, the optimization computation may be terminated when $\|x_{k+1}^s - x_k^s\| \le \varepsilon$, or $|J(x_k^s) - J(x_{k+1}^s)| \le \varepsilon$, where $\varepsilon > 0$ is a controlling parameter. More sophisticated stopping criteria may also be considered.

Since most optimization algorithms are iterative, there has been a fundamental trade-off between their efficiency and robustness [14]. In general, algorithms designed to be very efficient on one type of problem tend to be brittle in the sense that they may not be ideally used for other types of problem. Such a lack of a universally best algorithm is a manifestation of the so-called *No Free Lunch* (*NFL*) *theorems* [16]. The NFL theorems serve as a fundamental barrier to exaggerated claims of the power and efficiency of any specific algorithm in numerical optimizations. A way to cope with negative implications of the barrier is to restrict an algorithm to a particular class of problems and to design the algorithm structures only for the anticipated class. This has become a general principle in optimization.

2.2 Unconstrained Optimization

Let $J : \mathbb{R}^n \to \mathbb{R}$ be a sufficiently smooth objective function. Consider

$$y^* = \min_{x \in \mathbb{R}^n} J(x). \tag{2.2}$$

The above function J is referred as an *objective function*. We do not consider the maximization optimization problem due to the fact that $\max J(x) = -\min(-J(x))$, for $x \in S \subseteq \mathbb{R}^n$. The existence of a global minimizer for (2.2) has been shown in cases where the level sets of J are compact for certain γ [12].

2.2.1 *Optimality Conditions*

Theorem 2.2.1 (First-order Necessary Conditions, [11]) *If x^* is a local minimizer and J is continuously differentiable in an open neighborhood of x^*, then* $\nabla J(x^*) = 0$.

Theorem 2.2.2 (Second-order Necessary Conditions, [11]) *If x^* is a local minimizer and $\nabla^2 J$ is continuous in an open neighborhood of x^*, then $\nabla J(x^*) = 0$ and $\nabla^2 J(x^*)$ is positive semi-definite.*

Theorem 2.2.3 (Second-order Sufficient Conditions, [11]) *If $\nabla J(x^*) = 0$, $\nabla^2 J$ is continuous in an open neighborhood of x^* and $\nabla^2 J(x^*)$ is positive definite, then x^* is a strict local minimizer of J.*

Theorem 2.2.4 [11] *If J is convex, then any local minimizer x^* is a global minimizer of J. If in addition J is differentiable, then any stationary point x^* is a global minimizer of J.*

The above conditions provide a basis for the developments and analysis of various algorithms. In particular, any well-defined algorithm should verify whether putative solutions satisfy certain optimality conditions, and detect if a minimizer has been satisfactorily approximated. To determine a global minimizer of a given problem is in general difficult, therefore many algorithms used can only guarantee the convergence to a stationary point. The optimality and algorithms for constrained optimization are based on the results of unconstrained optimization, and additional techniques such as penalty methods and barrier methods are used to address equality and inequality constraints [2, 3, 11].

2.2.2 Line Search Methods

Each iteration in a line search method starts from x_k^s, computes a *search direction* p_k, and then decides *how far* to move along that direction. This iterative process can be illustrated by

$$x_{k+1}^s = x_k^s + \alpha_k p_k,$$

where the positive scalar α_k is called the *step length*. Most line search methods require p_k to be a *descent direction*, that is, $p_k^\top \nabla J(x_k^s) < 0$, to guarantee that the objective function value is reduced along that direction if the step length is sufficiently small. This is understood by using Taylor's theorem, which offers $J(x_k^s + \alpha_k p_k) = J(x_k^s) + \alpha_k p_k^\top \nabla J(x_k^s) + O(\alpha_k^2)$. Since the term $\alpha_k p_k^\top \nabla J(x_k^s)$ dominates $O(\alpha_k^2)$ for small α_k, it follows that $J(x_k^s + \alpha_k p_k) < J(x_k^s)$ for all positive but sufficiently small α_k if $p_k^\top \nabla J(x_k^s) < 0$. The *steepest-descent direction*,

$$p_k = -\nabla J\big(x_k^s\big),$$

is the most obvious choice for search direction. It is chosen among all the directions we could select from x_k^s, and it ensures that J decreases most rapidly.

According to Taylor's theorem, the rate of change in J along p_k at x_k^s is $p_k^\top \nabla J(x_k^s)$. Thus, if the condition $\|p_k\| = 1$ is imposed, the most rapid decrease is given by the solution of the problem

$$\min_{p_k}\big(p_k^\top \nabla J\big(x_k^s\big)\big).$$

Its solution can be found to be $p_k = -\nabla J(x_k^s)/\|\nabla J(x_k^s)\|$, which yields $p_k^\top \nabla J(x_k^s) = -\|\nabla J(x_k^s)\|$.

Other frequently used search directions include the *Newton direction*,

$$p_k = -\big(\nabla^2 J\big(x_k^s\big)\big)^{-1} \nabla J\big(x_k^s\big),$$

which can be adopted in a line search method when the Hessian matrix is positive definite. The basic idea here is to minimize

$$J_k(x) = J\big(x_k^s\big) + \big(x - x_k^s\big)^\top \nabla J\big(x_k^s\big) + \frac{1}{2}\big(x - x_k^s\big)^\top \nabla^2 J\big(x_k^s\big)\big(x - x_k^s\big),$$

the quadratic approximation of J at x_k^s instead of the objective function J itself. Setting the derivative of $J_k(x)$ equal to zero, one obtains

$$\nabla J\big(x_k^s\big) + \nabla^2 J\big(x_k^s\big)\big(x - x_k^s\big) = 0.$$

Therefore $p_k^\top \nabla J(x_k^s) = -p_k^\top \nabla^2 J(x_k^s) p_k \le -\sigma_k \|p_k\|$ for some $\sigma_k > 0$. Unless the gradient $\nabla J(x_k^s)$ is zero, $p_k^\top \nabla J(x_k^s) < 0$. Therefore, a Newton direction is a descent direction and a normalized unit step length is often utilized.

Needless to mention that calculations of the Hessian matrix may involve large amounts of computation. A *quasi-Newton method* is designed to avoid this disadvantage via features of $J(x_k^s)$ and $\nabla J(x_k^s)$. Their curvature information is used to construct the matrix B_k, an approximation of the Hessian matrix. The standard Quasi-Newton search routine is

$$p_k = -B_k^{-1} \nabla J\left(x_k^s\right).$$

A popular formula for obtaining B_k is the BFGS formula, named after Broyden, Fletcher, Goldfarb, and Shanno:

$$B_k = B_{k-1} - \frac{B_{k-1}^\top s_{k-1}^\top s_{k-1} B_{k-1}}{s_{k-1}^\top B_{k-1} s_{k-1}} + \frac{y_{k-1} y_{k-1}^\top}{y_{k-1}^\top s_{k-1}},$$

where $B_0 = I$, $s_{k-1} = x_k^s - x_{k-1}^s$ and $y_{k-1} = \nabla J(x_k^s) - \nabla J(x_{k-1}^s)$. Factorizations of B_k can be achieved through updating the inverse of B_{k-1} [11].

As yet another alternative, a *conjugate gradient direction* is computed by

$$p_k = -\nabla J\left(x_k^s\right) + \beta_k p_{k-1},$$

where $p_0 = -\nabla J(x_0^s)$, and β_k can either by computed via the Fletcher–Reeves formula,

$$\beta_k = \frac{\nabla J^\top (x_k^s) \nabla J(x_k^s)}{\nabla J^\top (x_{k-1}^s) \nabla J(x_{k-1}^s)},$$

or the Dixon formula,

$$\beta_k = -\frac{\nabla J^\top (x_k^s) \nabla J(x_k^s)}{p_{k-1}^\top \nabla J(x_{k-1}^s)}.$$

The Dixon formula β_k ensures p_k and p_{k+1} are conjugate, a concept originally developed for solutions of linear systems.

2.2.2.1 Step Length

Typical step-length selection algorithms consist of two phases: a *bracketing phase* and a *selection phase*. The former finds an interval $[a, b]$ containing acceptable step lengths, while the latter zooms in the interval to locate the final step length.

The second phase can be implemented by approximating solutions of the following scalar minimization problem:

$$\min_{\alpha>0} \phi(\alpha) = \min_{\alpha>0} J\left(x_k^s + \alpha p_k\right), \quad \alpha > 0, \tag{2.3}$$

which brings the name "line search." An exact solution for the above is called exact line search, which is expensive and frequently not necessary. More practical strategies suggest an *inexact line search* to determine a step size that makes an adequate reduction in J at minimal costs. To achieve this, often used conditions include

$$J\left(x_k^s + \alpha_k p_k\right) \le J\left(x_k^s\right) + c_1 \alpha_k p_k^\top \nabla J\left(x_k^s\right), \tag{2.4}$$

which prevents steps that are too long via a sufficient decrease criterion, and

$$p_k^\top \nabla J\left(x_k^s + \alpha_k p_k\right) \ge c_2 p_k^\top \nabla J\left(x_k^s\right), \tag{2.5}$$

which prevents steps that are too short via a curvature criterion, for $0 < c_1 < c_2 < 1$. Condition (2.4) is sometimes called the *Armijo condition,* while (2.5) is called the *Wolfe condition*. Moreover, in order to avoid poor choices of descent directions, an *angle condition* [9] can be introduced to enforce a uniformly lower bound on the angle θ_k between p_k and $-\nabla J(x_k^s)$:

$$\cos\theta_k = \frac{-p_k^\top \nabla J(x_k^s)}{\|p_k\| \|\nabla J(x_k^s)\|} \ge c_3 > 0, \tag{2.6}$$

where c_3 is independent of k. The above holds naturally in the method of steepest descent.

2.2.2.2 Convergence and Rate of Convergence

Definition 2.2.5 (First-order Convergence) First-order convergence of an optimization algorithm means that one (or some, or all) of the limit points of the iterate sequence is a stationary point of $J(x)$.

A standard first-order global convergence result for line search methods is

Theorem 2.2.6 [9] *Let $J : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and bounded from below. Further, let ∇J be Lipschitz continuous with constant $L > 0$, that is,*

$$\left\|\nabla J(y) - \nabla J(x)\right\| \le L\|y - x\| \quad \textit{for all } x, y \in \mathbb{R}^n.$$

If the sequence $\{x_k^s\}$ satisfies conditions (2.4), (2.5) *and* (2.6), *then*

$$\lim_{k\to\infty} \left\|\nabla J\left(x_k^s\right)\right\| = 0.$$

We can relax the assumptions in this theorem, where instead of requiring J to be bounded from below and continuously differentiable on $\mathbb{R}^n$, we only do so within an open set $\mathcal{N}$ containing the level set $\{x | J(x) \le J(x_0^s)\}$, where x_0^s is the starting point of the iteration. And the gradient ∇J is only required to be Lipschitz continuous on $\mathcal{N}$ [11].

Furthermore, the following theorem shows the linear convergence rate of the steepest-descent algorithm.

Theorem 2.2.7 [7] *Let $J : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable, and assume the Hessian matrix is positive definite. If the sequence $\{x_k^s\}$ is generated by a steepest-descent method with exact line search and it converges to x^*, then*

$$J\left(x_{k+1}^s\right) - J(x^*) \le \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)\left[J\left(x_k^s\right) - J\left(x^*\right)\right],$$

where $0 < \lambda_1 \le \cdots \le \lambda_n$ are the eigenvalues of the Hessian matrix of J.

It has been shown that numerical methods using Newton directions have a fast rate of local convergence, typically quadratic. Their main drawback, however, is the need of the Hessian matrix. There have been numerous recent discussions about the simplification of the underlying computation procedures. In the particularly practical case of the quasi-Newton method, if its search direction approximates the Newton direction accurately enough, then the unit step length can satisfy the Wolfe conditions as the iterates converge to a minimizer. Further, if for the search direction it holds that $\lim_{k\to\infty} \|\nabla J(x_k^s) + \nabla^2 J(x_k^s) p_k\| / \|p_k\| = 0$, then the quasi-Newton method offers a superlinearly convergent iteration. It is also known that for any quadratic objective function, a conjugate gradient method terminates with an optimal solution within n steps.

The following lemma will be used in the robustness analysis for line search methods.

Lemma 2.2.8 (Descent Lemma [2]) *Let $J : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable on $\mathbb{R}^n$. Suppose that ∇J is Lipschitz continuous with constant L. Then for $x, y \in \mathbb{R}^n$,*

$$J(x+y) \le J(x) + y^\top \nabla J(x) + \frac{L}{2}\|y\|^2.$$

Lemma 2.2.9 *Let $J : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable on $\mathbb{R}^n$. Suppose that ∇J is Lipschitz continuous with constant L. Let α_k, p_k be the step length and descent direction. Then*

$$J\left(x_k^s + \alpha_k p_k\right) - J\left(x_k^s\right) \le -\frac{c}{2L}\left\|\nabla J\left(x_k^s\right)\right\|^2 \cos^2\theta_k,$$

where $c = 1$ for exact line search, and $c = 2c_1(1 - c_2)$ for inexact line search satisfying conditions (2.4) and (2.5), and θ_k represents the angle between vector p_k and $-\nabla J(x_k^s)$.

Proof First, for exact line search, α_k is the solution of (2.3). From the Descent Lemma 2.2.8, we have $J(x_k^s + \alpha p_k) \le J(x_k^s) + \alpha p_k^\top \nabla J(x_k^s) + \frac{\alpha^2}{2} L\|p_k\|^2$ valid for all $\alpha > 0$. Letting $\bar{\alpha} = -\frac{p_k^\top \nabla J(x_k^s)}{L\|p_k\|^2} > 0$, it follows that

$$\begin{aligned} J\left(x_k^s + \alpha_k p_k\right) - J\left(x_k^s\right) &\le J\left(x_k^s + \bar{\alpha} p_k\right) - J\left(x_k^s\right) \quad \text{(exact line search)} \\ &\le \bar{\alpha} p_k^\top \nabla J\left(x_k^s\right) + \frac{\bar{\alpha}^2}{2} L\|p_k\|^2 \quad \text{(Descent Lemma 2.2.8)} \end{aligned}$$

$$= -\frac{(p_k^\top \nabla J(x_k^s))^2}{L\|p_k\|^2} + \frac{L\|p_k\|^2}{2}\frac{(p_k^\top \nabla J(x_k^s))^2}{(L\|p_k\|^2)^2}$$
$$= -\frac{1}{2L}\big\|\nabla J\big(x_k^s\big)\big\|^2 \cos^2\theta_k.$$

Second, for inexact line search, α_k satisfies conditions (2.4) and (2.5). From the Lipschitz condition, we have $p_k^\top[\nabla J(x_k^s+\alpha_k p_k)-\nabla J(x_k^s)] \le \|p_k\|\|\nabla J(x_k^s+\alpha_k p_k) - \nabla J(x_k^s)\| \le \alpha_k L\|p_k\|^2$. Then from (2.5), we have $-\alpha_k L\|p_k\|^2 \le p_k^\top[\nabla J(x_k^s) - \nabla J(x_k^s + \alpha_k p_k)] \le (1-c_2)p_k^\top \nabla J(x_k^s)$. That is, $-\alpha_k\|p_k\| \le -\frac{1-c_2}{L}\|\nabla J(x_k^s)\|\cos\theta_k$. Finally, from (2.4),

$$\begin{aligned} J\big(x_k^s+\alpha_k p_k\big) - J\big(x_k^s\big) &\le c_1\alpha_k p_k^\top \nabla J\big(x_k^s\big) \\ &= -c_1\alpha_k\|p_k\|\big\|\nabla J\big(x_k^s\big)\big\|\cos\theta_k \\ &\le -\frac{c}{2L}\big\|\nabla J\big(x_k^s\big)\big\|^2\cos^2\theta_k, \end{aligned}$$

where $c = 2c_1(1-c_2)$. □

Since $0 < c_1 < c_2 < 1$ is required to ensure the feasibility of inexact line search, we will have $c = 2c_1(1-c_2) < 1$. This observation is consistent with the upper bound results in the above lemma. That is, we always expect that the exact line search achieves more decrease along the search direction than the inexact line search.

2.2.2.3 Example: Minimization of the Rosenbrock's Function with Line Search Method

The Rosenbrock's function,

$$J(x) = 100\big(x_2 - x_1^2\big)^2 + (1-x_1)^2, \quad x \in \mathbb{R}^2, \tag{2.7}$$

also known as the "banana function," is a benchmark function in unconstrained optimization due to its curvature bends around the origin. The only global minimizer occurs at $x^* = [1, 1]^\top$, where $J(x^*) = 0$. A sequence $\{x_k^s\}$ obtained via the steepest-descent method with inexact line search staring from $x_0^s = [-1.9, 0]^\top$ is shown in Fig. 2.1, where the Armijo condition (2.4) is used and $c_1 = 0.4$. Due to the very slow curvature change of the banana function inside its "valley," the steepest-descent algorithm takes more than one thousand steps to converge.

2.2.3 *Trust-Region Methods*

At each iteration of a trust-region method, we consider the minimization of a *model function* m_k instead of the objective function J at the current iterate x_k^s. Because the

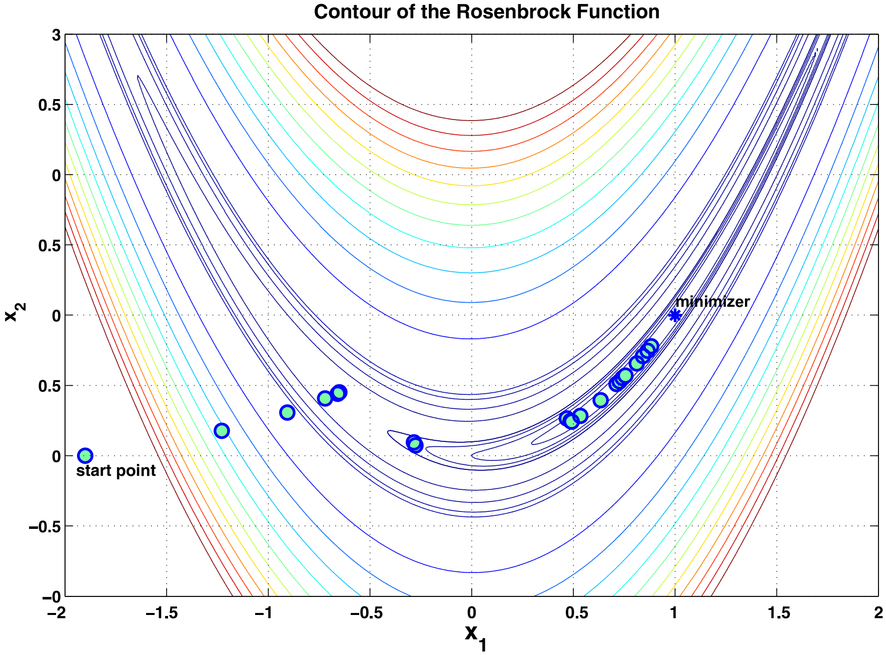


Fig. 2.1 Steepest-descent method on Rosenbrock's function

model function may not be a good approximation of J when x is far away from x_k^s, we have to restrict the search for a minimizer of m_k to a local region around x_k^s. Such a region is called a *trust region*. A trust-region method is defined as

$$\min_{\|p\| \leq \Delta_k} m_k\big(x_k^s + p\big). \tag{2.8}$$

Let p_k be the minimizer obtained, and Δ_k the current size of the trust region. The current iterate is then updated to be $x_k^s + p_k$. If the achieved objective function reduction is sufficient compared with the reduction predicted by the model, the trial point is accepted as the new iterate and the trust region is centered at the new point and possibly enlarged. On the other hand, if the achieved reduction is poor compared with the predicted one, the current iterate is typically left unchanged and the trust region is reduced. This process is then repeated until convergence occurs.

Define the ratio

$$\rho_k = \frac{J(x_k^s) - J(x_k^s + p_k)}{m_k(x_k^s) - m_k(x_k^s + p_k)}. \tag{2.9}$$

The following algorithm [11] describes the process.

2.2.3.1 Trust-Region Algorithm

Step 0 Given $\bar{\Delta} > 0$, initialize the trust-region size to $\Delta_0 \in (0, \bar{\Delta})$, and $\eta \in [0, \frac{1}{4})$. Set $k = 0$.
Step 1 Approximately solve the trust-region problem (2.8) to obtain p_k.
Step 2 Evaluate ρ_k from (2.9).
Step 3 If $\rho_k < \frac{1}{4}$, $\Delta_{k+1} = \frac{1}{4}\|p_k\|$; if $\rho > \frac{3}{4}$ and $\|p_k\| = \Delta_k$, $\Delta_{k+1} = \min(2\Delta_k, \bar{\Delta})$; else $\Delta_{k+1} = \Delta_k$.
Step 4 If $\rho_k > \eta$, $x^s_{k+1} = x^s_k + p_k$, else $x^s_{k+1} = x^s_k$. Set $k = k + 1$. Go to Step 1.

Quadratic approximations of J are often used for constructing m_k. In this case, m_k in (2.8) can be formed as

$$m_k(p) = J\big(x^s_k\big) + g_k^\top p + \frac{1}{2} p^\top B_k p. \tag{2.10}$$

The vector g_k is either the gradient $\nabla J(x^s_k)$ or an approximation of it, and the matrix B_k is either the Hessian matrix $\nabla^2 J(x^s_k)$ or an approximation of it. Thus, such construction of m_k still requires gradient information. However, the trust-region framework provides large flexibility in designing derivative-free optimization methods. This compares very favorable with most line search methods which do require gradient measurements of the objective function. Derivative-free trust-region algorithms proposed in [4, 5, 13] use multivariate interpolation to construct the model function m_k, where only an interpolation set Y containing the interpolating nodes and their objective function values are needed. Overall, trust-region methods retain the quadratic convergence rate while being globally convergent. The following is a global convergence result for trust-region methods [11].

Theorem 2.2.10 *Let $J : \mathbb{R}^n \to \mathbb{R}$ be Lipschitz, continuously differentiable and bounded below on the level set $\{x \in \mathbb{R}^n | J(x) \le J(x^s_0)\}$. Further, let $\eta > 0$ in the trust-region algorithm. Suppose that $\|B_k\| \le \beta$ for some constant β, and that all approximate solutions of* (2.8) *satisfy the inequality*

$$m_k(0) - m_k(p_k) \ge c_t \big\|\nabla J\big(x^s_k\big)\big\| \min\left(\Delta_k, \frac{\|\nabla J(x^s_k)\|}{\|B_k\|}\right)$$

for some constant $c_t \in (0, 1]$, and $\|p_k\| \le \gamma \Delta_k$ for some constant $\gamma \ge 1$. Then

$$\lim_{k\to\infty} \big\|\nabla J\big(x^s_k\big)\big\| = 0.$$

2.2.3.2 Example: Minimization of the Rosenbrock's Function with Trust-Region Method

Again we use the banana function to illustrate the trust-region method. A sequence $\{x^s_k\}$ obtained via the trust-region method starting from $x^s_0 = [-1.9, 0]^\top$ is shown in Fig. 2.2, where a quadratic approximation (2.10) is used and the measurements of exact gradient and Hessian are assumed. The trust region $\Delta_0 = 0.5$ and the iterates converge to x^* in 18 steps.

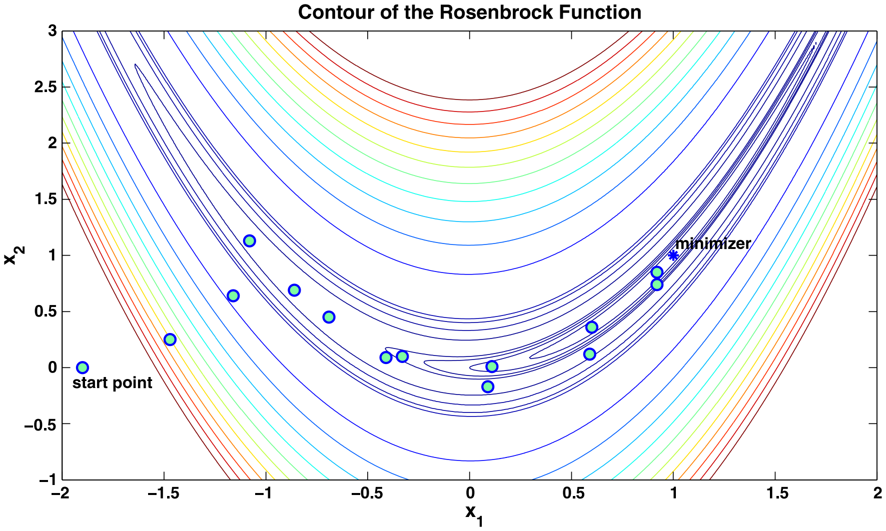


Fig. 2.2 Trust-region method on Rosenbrock's function

2.2.4 Direct Search Methods

Direct search methods are one of the best known methods within the family of derivative-free unconstrained optimization. In the past decades, these methods have seen a revival of interest due to the appearance of rigorous mathematical analysis [1, 15], as well as in parallel and distributed computing. Such features make direct search applicable to the problem of extremum seeking control design. And as direct search does not need derivative information, it can apply to non-smooth objective functions as well. Overall, direct search methods are slower than line search methods, such as steepest-descent method. A systematic review of direct search methods can be found in [9].

The well-known Simplex algorithm of Nelder and Mead [10] is one of the direct search methods. Compass search is one of the earlier version of two dimensional direct search, and it can be summarized as follows: Try steps to the East, West, North, and South. If one of these steps yields a reduction in the function, the improved point becomes the new iterate. If none of these steps yields improvement, try again with steps half as long. By revisiting compass search in a more analytically rigorous manner, it has been named "generating set search" or "pattern search method" [9].

2.2.4.1 Generating Set Search Algorithm

Step 0 Let x_0 be the initial guess. Set $\Delta_{tol} > 0$ as the tolerance used for convergence, and let $\Delta_0 > \Delta_{tol}$ be the initial value of the step-length control parameter.

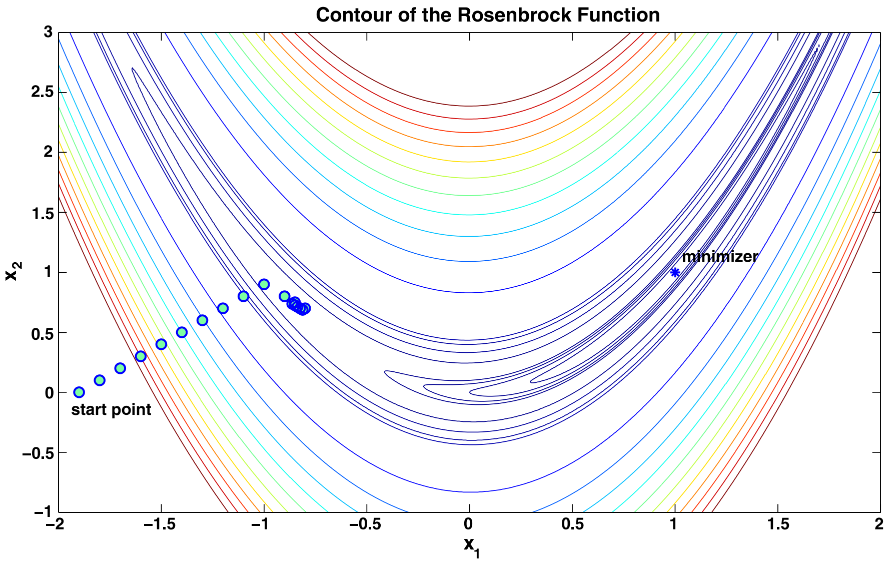


Fig. 2.3 Direct search method on Rosenbrock's function

Step 1 Let D_s be the coordinate direction set (or generator set)

$$D_s = \{e_1, e_2, \dots, e_n, -e_1, -e_2, \dots, -e_n\},$$

where e_i is the ith unit coordinate vector in $\mathbb{R}^n$.

Step 2 If there exists $d_k \in D_s$ such that $f(x_k + \Delta_k d_k) < f(x_k)$, then set $x_{k+1} = x_k + \Delta_k d_k$ and $\Delta_{k+1} = \Delta_k$. Set $k = k + 1$ and go to Step 1.

Step 3 Otherwise, if $f(x_k + \Delta_k d) \geq f(x_k)$ for all $d \in D_s$, set $x_{k+1} = x_k$ and $\Delta_{k+1} = \Delta_k/2$. If $\Delta_{k+1} < \Delta_{tol}$, then terminate; otherwise set $k = k + 1$ and go to Step 1.

As depicted for two dimensional compass search, it is easy to see that at each iteration, at least one of the four coordinate directions will be a descent direction. In fact, it is true for any dimension n: given any $x \in \mathbb{R}^n$ for which $\Delta f(x) \neq 0$, at least one of coordinate directions must be a descent direction.

We choose the generator set D_s as $\{e_1, e_2, \dots, e_n, -e_1, -e_2, \dots, -e_n\}$ in the above algorithm. In general, it can be any positive spanning set [6]. That is, for n dimensional optimization problem, the minimum number of vectors in the generator set is $n + 1$, which will guarantee a descent direction can be found in the generator set.

Direct search can be thought of as being related to trust-region methods, although in direct search no attempt is done to approximate the objective function nor its gradient, as trust-region methods do. Thus, direct search methods are best suited to problems for which no derivative information is available; in particular, to problems where the objective function is non-smooth.

2.2.4.2 Example: Minimization of the Rosenbrock's Function with Direct Search Method

Again we use the banana function to illustrate the direct search method. A sequence $\{x_k^s\}$ is obtained via the direct search method, starting from $x_0^s = [-1.9, 0]^\top$. The resulting sequence is shown in Fig. 2.3, where the generator set

$$D_s = \{(1, 1), (1, -1), (-1, 1), (-1, 1)\}$$

is used and no derivative information is employed. The initial step length is $\Delta_0 = 0.1$. Only the first 20 steps of the simulation are shown, where it can be seen in Fig. 2.3 that the sequence does converge to the neighborhood of the minimum, but at a very slow rate due to the small gradient change near the minimum, which is located inside an almost flat "valley."

References

1. Audet, C., Dennis, J.E.: Analysis of generalized pattern searches. SIAM J. Optim. **13**(3), 889–903 (2002)
2. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, Belmont (1995)
3. Blowey, J.F., Craig, A.W., Shardlow, T.: Frontiers in Numerical Analysis. Springer, Berlin (2002)
4. Conn, A.R., Scheinberg, K., Toint, P.L.: On the convergence of derivative-free methods for unconstrained optimization. Approximation Theory and Optimization: Tributes to M.J.D. Powell, pp. 83–108 (1997)
5. Conn, A.R., Scheinberg, K., Toint, P.L.: Recent progress in unconstrained nonlinear optimization without derivatives. Math. Program. **79**, 397–414 (1997)
6. Davis, C.: Theory of positive linear dependence. Am. J. Math. **76**(4), 733–746 (1954)
7. Fletcher, R.: Practical Methods of Optimization. Wiley, New York (1987)
8. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (1985)
9. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev. **45**(3), 385–482 (2003)
10. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**(4), 308–313 (1965)
11. Nocedal, J., Wright, S.: Numerical Optimization. Springer, Berlin (1999)
12. Pardalos, P.M., Resende, M.G.C.: Handbook of Applied Optimization. Oxford University Press, New York (2002)
13. Powell, M.J.D.: UOBYQA: Unconstrained optimization by quadratic approximation. Technical Report NA2000/14, DAMTP. University of Cambridge (2000)
14. Spall, J.C.: Introduction to Stochastic Search and Optimization. Wiley-Interscience, New Jersey (2003)
15. Torczon, V.: On the convergence of pattern search algorithms. SIAM J. Optim. **7**(1), 1–25 (1997)
16. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**, 67–82 (1997)

Chapter 3
Design of Extremum Seeking Control

Throughout this chapter, we will deal with the general nonlinear system

$$\dot{x} = f(x, u), \tag{3.1}$$

$$y = J(x), \tag{3.2}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}$ is the input, $y \in \mathbb{R}$ is the *performance output*, and the functions $f : D \times \mathbb{R} \to \mathbb{R}^n$ and $J : D \to \mathbb{R}$ are sufficiently smooth on $D \subseteq \mathbb{R}^n$. For the simplification of analysis, we assume $D = \mathbb{R}^n$ throughout the book unless otherwise stated. We consider the design of an extremum seeking controller to find the extremum (maximum or minimum) of the performance function (3.2). The reader should note that a maximum seeking controller design can be used to achieve minimum seeking by replacing y with $-y$, and vice versa.

The performance output $y = J(x)$ should not be confused with the common notion of a plant output, which is generally used for regulation or tracking purposes. Here, the performance output is the aspect of plant behavior one desires to minimize or maximize, and it may or may not be equal to (or a function of) a physical output obtained via sensor measurements. In other words, the performance output is application and design dependent.

3.1 Analog Optimization Based Extremum Seeking Control

In this section, we focus on maximum seeking control, where the nonlinearity with an extremum arises as a reference-to-output equilibrium map of a general nonlinear system [2]. This system is assumed to be stable or stabilizable at each of these equilibria by a local feedback controller. Suppose that we know a smooth control law

$$u = \alpha(x, \theta) \tag{3.3}$$

parameterized by a scalar parameter θ. The closed-loop system

$$\dot{x} = f\bigl(x, \alpha(x, \theta)\bigr)$$

C. Zhang, R. Ordóñez, *Extremum-Seeking Control and Applications*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2224-1_3, © Springer-Verlag London Limited 2012

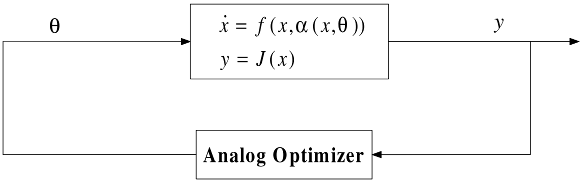


Fig. 3.1 Analog optimization based extremum seeking control

then has equilibria parameterized by θ. We make the following assumptions about the closed-loop system.

Assumption 3.1.1 *There exists a smooth function $l : \mathbb{R} \to \mathbb{R}^n$ such that*

$$f\big(x, \alpha(x,\theta)\big) = 0 \quad \textit{if and only if } x = l(\theta).$$

Assumption 3.1.2 *For each $\theta \in \mathbb{R}$, the equilibrium $x = l(\theta)$ of the system (3.1) is locally exponentially stable.*

Assumption 3.1.3 *There exists $\theta^* \in \mathbb{R}$ such that*

$$(J \circ l)'\big(\theta^*\big) = 0,$$
$$(J \circ l)''\big(\theta^*\big) < 0.$$

Thus, we assume that the output equilibrium map

$$y = J\big(l(\theta)\big) \tag{3.4}$$

has a maximum at $\theta = \theta^*$ (we would assume $(J \circ l)''(\theta^*) > 0$ for a minimization problem). Our objective is to develop a feedback mechanism that maximizes the steady state value of y but without requiring knowledge of either θ^* or the functions J and l.

The above three assumptions were first proposed in [19]. Using these assumptions, one reduces the n-dimensional optimization of the performance function (3.2) to the one-dimensional problem of optimizing (3.4) in the steady state. Therefore, the design of extremum seeking control focuses on how to find an optimizing law for the parameter θ, where several interesting analog optimizers come into the context of the extremum seeking. This framework allows the AOESC approach to be considered model-free, because Assumption 3.1.2 means that we have a control law designed for local stabilization and this control law need not to be based on modeling knowledge of either $f(x,u)$ or $l(\theta)$. In practice, however, one may usually find that this knowledge is indeed required in order to devise the local control law (3.3). A basic block diagram depicting the AOESC scheme can be found in Fig. 3.1.

3.1.1 Gradient Based Extremum Seeking Control

Consider the maximization of the performance function $y = J(\theta)$, where $\theta \in \mathbb{R}$. If we know the derivative $dJ/d\theta$, we can choose the optimizing law for θ as

$$\dot{\theta} = k\frac{dJ}{d\theta}, \quad k > 0. \tag{3.5}$$

By letting θ^* be an isolated local maximizer of $J(\theta)$, we can choose the Lyapunov candidate $V = J(\theta^*) - J(\theta)$. Then,

$$\dot{V} = -\frac{dJ}{d\theta}\dot{\theta} = -k\left(\frac{dJ}{d\theta}\right)^2 \leq 0.$$

Thus, we see that θ converges to the invariant set where $\dot{V} = 0$; that is, where $dJ/d\theta = 0$, which can only occur at $\theta = \theta^*$ for a local region (due to Assumption 3.1.3). Therefore, by invoking the invariance principle [17], we conclude that the optimizing law (3.5) can successfully maximize $J(\theta)$. Note that we can easily change it into a minimizing law by changing the sign of V and k.

However, assuming the knowledge of the derivative amounts to knowing $J(\theta)$ as well, and we can obtain θ^* by solving the equation $dJ/d\theta = 0$, which also means that the root finding and optimization problem are in this sense mathematically equivalent.

In the one-dimensional case, estimating the derivative is not a difficult task, and therefore we do not really need the precise gradient information. In fact, one can choose

$$\theta = k\,\mathrm{sgn}\left(\frac{dJ}{d\theta}\right), \tag{3.6}$$

with $k > 0$, where sgn is the signum function, defined as

$$\mathrm{sgn}(t) = \begin{cases} 1, & t \geq 0 \\ -1, & t < 0 \end{cases}.$$

Then, using the same Lyapunov candidate $V = J(\theta^*) - J(\theta)$, we have

$$\dot{V} = -k\frac{dJ}{d\theta}\,\mathrm{sgn}\left(\frac{dJ}{d\theta}\right) = -k\left|\frac{dJ}{d\theta}\right| \leq 0.$$

Assuming knowledge of the sign of the derivative is still a strong assumption. However, we can instead try to estimate the sign, which is exactly what the sliding mode approach tries to achieve. One can then form a gradient based extremum seeking control as the framework shown in Fig. 3.1 by using the update law (3.6) to be the analog optimizer.

Below we will see that a general design of extremum seeking control based on gradient feedback is difficult without reducing it to the framework in Fig. 3.1. Now, consider a general n-dimensional gradient system

$$\dot{x} = k\nabla J(x), \quad k > 0. \tag{3.7}$$

It is well known [15, 16] that the maximal points of J are stable equilibria of the gradient system (3.7), and that, if the level sets of J are bounded, then the trajectory of x will converge asymptotically to the set of stationary points of J. So if we have the gradient $\nabla J(x)$ and if we can design a control law u to force the nonlinear system (3.1) with performance function (3.2) to behave as the gradient system (3.7), then we can achieve the extremum seeking control task.

Moreover, we can also try to form the system

$$\dot{x} = kp(x), \quad k > 0,$$

where $p(x)$ is required to be an ascent direction (that is $p(x)^\top \nabla J(x) > 0$) in order to guarantee that the performance function is increased along the direction of $\dot{x}$, which can be seen from $\dot{J} = \nabla J^\top \dot{x} = k\nabla J^\top p(x) > 0$. Recent developments in [7] show that we can achieve finite convergence to the stationary point by choosing a control law u to force the system to be

$$\dot{x} = \frac{\nabla J(x)}{\|\nabla J(x)\|}$$

or

$$\dot{x} = \operatorname{sgn}\bigl(\nabla J(x)\bigr).$$

Now, we want to see how to design the control u to force the dynamic system to behave like the gradient system (3.7). Consider a linear time invariant (LTI) system

$$\dot{x} = Ax + Bu, \tag{3.8}$$

where $x \in \mathbb{R}^n$. Assume that x^* is a local maximum of the performance function $J(x)$. Let $V = J(x^*) - J(x)$ be a Lyapunov candidate, then

$$\dot{V} = -\nabla J(x)^\top \dot{x} = -\nabla J(x)^\top (Ax + Bu).$$

We need to find a control law such that $\dot{V} \le 0$, which generally is very difficult even if we know the gradient. In the particular case when the LTI system is square, that is, $u \in \mathbb{R}^n$ and B is nonsingular, we can choose

$$u = kB^{-1}\bigl(\nabla J(x) - Ax\bigr),$$

with $k > 0$. Then, $\dot{V} = -k\|\nabla J(x)\|^2 \le 0$ and we can conclude the state will converge to the stationary points of J. Of course, we can have a more flexible design by choosing $u = kB^{-1}(p(x) - Ax)$, with $k > 0$ and $p(x)$ satisfying $p(x)^\top \nabla J(x) > 0$.

Recent research on gradient or its estimation based extremum seeking can be found in [1, 3–6, 11, 24]. However, the requirement of knowing the gradient $\nabla J(x)$ is a very strong assumption, and moreover for the general single input, single output

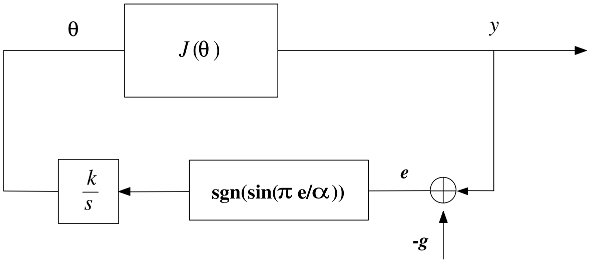


Fig. 3.2 Sliding mode based analog optimization

n-dimensional nonlinear system (3.1) and (3.2), a control law that transforms the nonlinear system into a gradient system (3.7) may not exist or may be very difficult to find. Thus, by posing Assumptions 3.1.1–3.1.3, we can reduce the extremum seeking control design to find a controller parameter update law for θ, which optimizes the unimodal reference-to-output equilibrium map $J \circ l(\theta)$. For these reasons, analog optimization laws based on these assumptions have been intensively explored.

3.1.2 Sliding Mode Based Extremum Seeking Control

The use of sliding mode for analog optimization of an analytically unknown one-dimensional function $J(\theta)$ has been reported in [12, 18, 26]. The basic idea is to make J follow an increasing/decreasing time function via sliding mode motions. The main difficulty with such set up is that the unknown gradient term multiplies the control at the differential equation of J so that the system itself possesses a variable structure behavior. This idea has been extended in [8] with the introduction of the notion of periodic switching function and then studied in [12, 13, 20–23, 25, 27, 28, 30, 31] on a variety of automotive problems, especially in ABS design [9, 10, 14, 29].

Consider the maximization of a performance function $y = J(\theta)$. The performance output y is forced to track an increasing time function irrespective of the unknown gradient via sliding mode. A basic sliding mode based analog optimization method can be found in Fig. 3.2, where the symbol $\bigoplus$ denotes a summer. Pick any increasing function $g(t)$ and try to keep $J(\theta) - g(t)$ at a constant value by a proper choice of $\dot{\theta}$. If so, $J(\theta)$ increases at the same rate as $g(t)$, independent of whether $\theta < \theta^*$ or $\theta > \theta^*$. To this end, let

$$e = J(\theta) - g(t), \tag{3.9}$$

so that

$$\dot{e} = \frac{dJ}{d\theta}\dot{\theta} - \dot{g}(t).$$

With the optimizing law of

$$\dot{\theta} = k\,\mathrm{sgn}(\sin(\pi e/\alpha)), \quad k > 0 \tag{3.10}$$

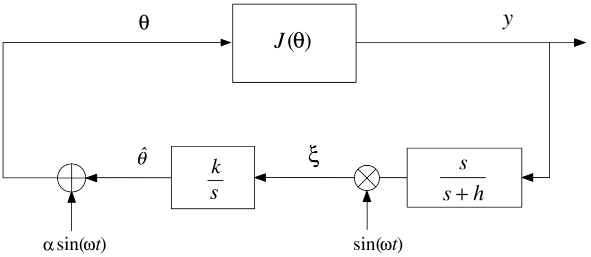


Fig. 3.3 Sinusoidal perturbation based analog optimization

as in [8] with α being a small positive constant, a sliding motion occurs for $k|dJ/d\theta| > |\dot{g}(t)|$ and θ is steered toward θ^* while y tracks $g(t)$. The region defined by $|dJ/d\theta| < |\dot{g}(t)|/k$ quantifies the region in which θ will be confined with this optimizing law. The idea can be extended to more general dynamics by adding the derivatives of the performance function as well as those of $g(t)$ to the sliding manifold expression so as to compensate the relative degree deficit. In [13], this optimization idea has further been developed for online operating point and set point optimization purposes by using a two-time scale sliding mode optimization design. The resulting method allows the optimization of the closed-loop operation of a system by exploiting the extra degree of freedom in the available control authority, possibly in a different time scale. This is exactly what we mean to be sliding mode based extremum seeking control.

3.1.3 Perturbation Based Extremum Seeking Control

The method of sinusoidal perturbation introduced in this section has been the most popular of extremum seeking control schemes. Due to its continuous way to perform gradient type optimization, it permits fast adaptation and easy implementation. This section is mainly based on Chap. 5 of [2], and later we extend the perturbation based extremum seeking control to unstable systems by incorporating phase lead compensators in the extremum seeking loop.

Consider the maximization of a performance function $y = J(\theta)$. A basic sinusoidal perturbation based continuous maximization method can be found in Fig. 3.3, where the symbol $\bigotimes$ denotes a multiplier.

The perturbation signal $\alpha \sin(\omega t)$ fed into the function helps to get a measure of the gradient information of $J(\theta)$. The following result [2] summarizes the properties of the basic perturbation based extremum seeking loop in Fig. 3.3:

Theorem 3.1.4 *For sufficiently large ω there exists a unique exponentially stable periodic solution of period $2\pi/\omega$ for system in Fig. 3.3 and it satisfies*

$$\left|\tilde{\theta}^{2\pi/\omega}(t)\right| + \left|e^{2\pi/\omega}(t) - \frac{\alpha^2 J''}{4}\right| \le O\left(\frac{1}{\omega}\right), \quad \forall t \ge 0.$$

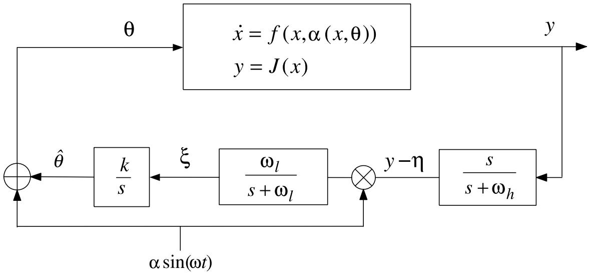


Fig. 3.4 Perturbation based extremum seeking control

This theorem states a local convergence property of the perturbation based continuous optimization of a single parameter function. The output $y - J^*$ converges to $J''O(\frac{1}{\omega^2} + \alpha^2)$. This convergence result is of second order, and the convergence speed is proportional to $1/\omega, \alpha, k$ and J''.

Based on the analog optimization scheme, the perturbation based extremum seeking control scheme is shown in Fig. 3.4. Here, we review the results that first appeared in [19], and later in Chap. 5 of [2]. These results lay the foundation of the techniques we use here to extend the perturbation based extremum seeking control to moderately unstable system and the autonomous vehicle source seeking problem.

Tools of averaging and singular perturbation are employed to show that solutions of the closed-loop system converge to a small neighborhood of the extremum of the equilibrium map. The size of the neighborhood is inversely proportional to the adaptation gain and the amplitude and frequency of a periodic signal used to achieve extremum seeking. The low pass filter $\frac{\omega_l}{s+\omega_l}$ is not necessary, but it is helpful in filtering out a $\cos(2\omega t)$ signal after the multiplier (demodulator). The design parameters are selected as

$$\omega_h = \omega\omega_H = \omega\delta\omega'_H = O(\omega\delta), \tag{3.11}$$

$$\omega_l = \omega\omega_L = \omega\delta\omega'_L = O(\omega\delta), \tag{3.12}$$

$$k = \omega K = \omega\delta K' = O(\omega\delta), \tag{3.13}$$

where ω and δ are small positive constants and ω'_H, ω'_L, and K' are $O(1)$ positive constants. As it will be become apparent later, α also needs to be small. From (3.11) and (3.12) we see that the cut-off frequencies of the filters need to be lower than the frequency of the perturbation signal, from which it follows that we need to choose $\omega_H < 1$ and $\omega_L < 1$. In addition, the adaptation gain k needs to be small as well.

The analysis that follows treats first the static case ("freeze" x at its equilibrium) using the method of averaging. Then we use the singular perturbation method for the full system in Fig. 3.4. Let us introduce the new coordinates $\tilde{\theta} = \hat{\theta} - \theta^*$ and $\tilde{\eta} = \eta - J \circ l(\theta^*)$, where $\dot{\eta} = -\omega_h\eta + \omega_h y$ from its definition in Fig. 3.4. Then, in the time scale $\tau = \omega t$, the system in Fig. 3.4 is written as

$$\omega\frac{dx}{d\tau} = f\big(x, \alpha(x, \theta^* + \tilde{\theta} + \alpha\sin\tau)\big), \tag{3.14}$$

$$\frac{d}{d\tau}\begin{bmatrix}\tilde{\theta}\\ \xi\\ \tilde{\eta}\end{bmatrix}=\delta\begin{bmatrix}K'\xi\\ -\omega'_L\xi+\omega'_L(J(x)-J\circ l(\theta^*)-\tilde{\eta})\alpha\sin\tau\\ -\omega'_H\tilde{\eta}+\omega'_H(J(x)-J\circ l(\theta^*))\end{bmatrix}. \tag{3.15}$$

Then we have the following theorem.

Theorem 3.1.5 [2] *Consider the feedback system* (3.14) *and* (3.15) *under Assumptions* 3.1.1–3.1.3. *There exists a ball of initial conditions around the point* $(x,\hat{\theta},\xi,\eta)=(l(\theta^*),\theta^*,0,J\circ l(\theta^*))$ *and contains* $\bar{\omega}$, $\bar{\delta}$, *and* $\bar{\alpha}$ *such that for all* $\omega\in(0,\bar{\omega})$, $\delta\in(0,\bar{\delta})$, *and* $\alpha\in(0,\bar{\alpha})$, *the solution* $(x(t),\hat{\theta}(t),\xi(t),\eta(t))$ *converges exponentially to an* $O(\omega+\delta+\alpha)$*-neighborhood of that point. Furthermore,* $y(t)$ *converges to an* $O(\omega+\delta+\alpha)$*-neighborhood of* $J\circ l(\theta^*)$.

3.1.4 Perturbation Based Extremum Seeking Control for a Plant with Slightly Unstable Poles

The perturbation based extremum seeking control above and in [2] relies on time scale decomposition and as such has so far been developed only for plants that are open loop stable, with poles that are sufficiently well damped. In the current and following sections, we introduce a new idea regarding how to extend the applicability of perturbation based extremum seeking to moderately unstable systems (this material is related to [32], where the focus is mainly on source seeking). The extension to marginally unstable systems draws motivation from the application of autonomous vehicle source seeking and will be presented in Chap. 8.

We present an example of a MIMO plant with slightly unstable poles that can be stabilized, in the absence of its output measurements, with extremum seeking. Consider a two-input–two-output system:

$$\begin{aligned}\dot{x}&=v_x+\varepsilon_x x,\\ \dot{y}&=v_y+\varepsilon_y y,\end{aligned} \tag{3.16}$$

with performance function

$$J=f(x,y)=f^*-q_x\left(x-x^*\right)^2-q_y\left(y-y^*\right)^2, \tag{3.17}$$

where $\varepsilon_x,\varepsilon_y>0$ are constant and v_x, v_y are the inputs. The (x^*,y^*) is a maximizer, $f^*=f(x^*,y^*)$ is the maximum and q_x, q_y are some unknown positive constants (since the Hessian is negative). General non-quadratic maps with non-diagonal Hessians are equally amenable to analysis, using the same technique as in [2, 19]. A block diagram of extremum seeking is shown in Fig. 3.5. If ε_x, ε_y are very small, the robustness of the perturbation based extremum seeking loop itself will be enough to compensate for their effect without having to resort to a phase lead compensator.

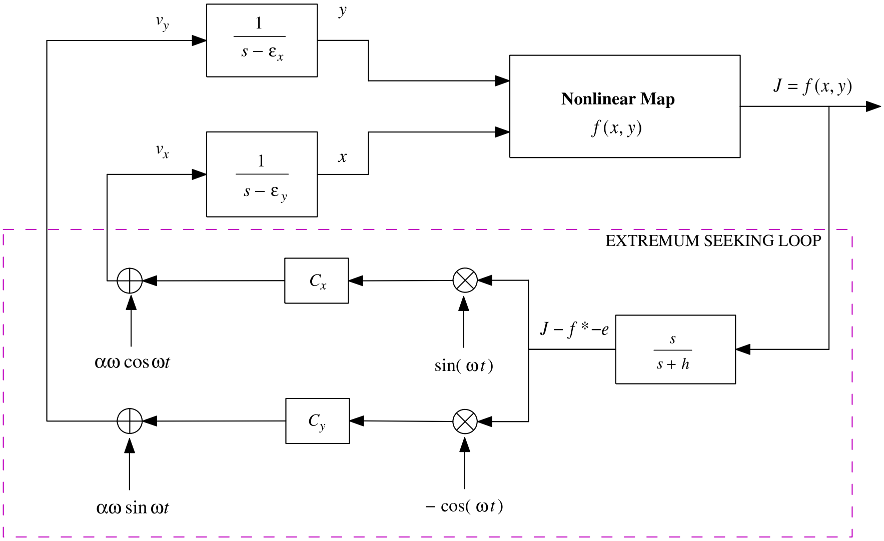


Fig. 3.5 Perturbation based extremum seeking control of a plant with slightly unstable poles

The analysis that follows employs the method of averaging. Let

$$e=\frac{h}{s+h}[J]-f^*, \tag{3.18}$$

then the signal after the washout filter can be expressed as

$$\frac{s}{s+h}[J]=J-\frac{h}{s+h}[J]=J-f^*-e.$$

Now, let us introduce the new coordinates

$$\tilde{x}=x-x^*-\alpha\sin(\omega t), \tag{3.19}$$

$$\tilde{y}=y-y^*+\alpha\cos(\omega t). \tag{3.20}$$

Then, in the time scale $\tau=\omega t$, we summarize the system in Fig. 3.5 as

$$\begin{aligned}\frac{d\tilde{x}}{d\tau}&=\frac{1}{\omega}\big[c_x\Delta\sin\tau+\varepsilon_x(\tilde{x}+x^*+\alpha\sin\tau)\big],\\ \frac{d\tilde{y}}{d\tau}&=\frac{1}{\omega}\big[-c_y\Delta\cos\tau+\varepsilon_y(\tilde{y}+y^*-\alpha\cos\tau)\big],\\ \frac{de}{d\tau}&=\frac{h}{\omega}\Delta,\end{aligned} \tag{3.21}$$

where Δ is defined as

$$\Delta=(J-f^*-e)=-\big[q_x(\tilde{x}+\alpha\sin\tau)^2+q_y(\tilde{y}-\alpha\cos\tau)^2+e\big]. \tag{3.22}$$

The system (3.21) is in a form to which the averaging method is applicable. The averaging model of (3.21) is

$$\begin{aligned}
\frac{d\tilde{x}_{avg}}{d\tau} &= \frac{1}{\omega}\big[(\varepsilon_x - \alpha c_x q_x)\tilde{x}_{avg} + \varepsilon_x x^*\big],\\
\frac{d\tilde{y}_{avg}}{d\tau} &= \frac{1}{\omega}\big[(\varepsilon_y - \alpha c_y q_y)\tilde{y}_{avg} + \varepsilon_y y^*\big],\\
\frac{de_{avg}}{d\tau} &= \frac{1}{\omega}(-h)\left[q_x\tilde{x}^2_{avg} + q_y\tilde{y}^2_{avg} + e_{avg} + \frac{\alpha^2}{2}(q_x+q_y)\right].
\end{aligned} \tag{3.23}$$

Then the equilibrium of the average model (3.23) is

$$\begin{aligned}
\tilde{x}^e_{avg} &= \frac{\varepsilon_x x^*}{\alpha c_x q_x - \varepsilon_x},\\
\tilde{y}^e_{avg} &= \frac{\varepsilon_y y^*}{\alpha c_y q_y - \varepsilon_y},\\
e^e_{avg} &= -\frac{\alpha^2}{2}(q_x+q_y) - q_x\left(\frac{\varepsilon_x x^*}{\alpha c_x q_x - \varepsilon_x}\right)^2 - q_y\left(\frac{\varepsilon_y y^*}{\alpha c_y q_y - \varepsilon_y}\right)^2.
\end{aligned} \tag{3.24}$$

The Jacobian of (3.23) at $(\tilde{x}^e_{avg}, \tilde{y}^e_{avg}, e^e_{avg})$ is

$$J_{avg} = \frac{1}{\omega}\begin{bmatrix} \varepsilon_x - \alpha c_x q_x & 0 & 0 \\ 0 & \varepsilon_y - \alpha c_y q_y & 0 \\ -2hq_x\tilde{x}^e_{avg} & -2hq_y\tilde{y}^e_{avg} & -h \end{bmatrix}. \tag{3.25}$$

Therefore, J_{avg} will be Hurwitz if and only if

$$\varepsilon_x - \alpha q_x c_x > 0, \qquad \varepsilon_y - \alpha q_y c_y > 0. \tag{3.26}$$

Given that q_x, q_y are unknown but positive constants, and that ε_x, ε_y are small, there exist some α, c_x and c_y satisfying the inequalities in (3.26), so as to make the Jacobian (3.25) Hurwitz. This implies that the equilibrium (3.24) of the average system (3.23) is exponentially stable. Then, according to the Averaging Theorem [17], we have the following result.

Theorem 3.1.6 *Consider the system in Fig. 3.5, where the nonlinear map has the form of (3.17). There exist $\bar{\varepsilon}, \bar{\omega}$ such that for all $\varepsilon_x, \varepsilon_y \in (0, \bar{\varepsilon})$ and for all $\frac{1}{\omega} \in (0, \frac{1}{\bar{\omega}})$ the system has a unique exponentially stable periodic solution $(\tilde{x}^{2\pi/\omega}, \tilde{y}^{2\pi/\omega}, e^{2\pi/\omega})$ of period $\frac{2\pi}{\omega}$, and this solution satisfies*

$$\left\|\begin{bmatrix} \tilde{x}^{2\pi/\omega} - \tilde{x}^e_{avg} \\ \tilde{y}^{2\pi/\omega} - \tilde{y}^e_{avg} \\ e^{2\pi/\omega} - e^e_{avg} \end{bmatrix}\right\| \le O(1/\bar{\omega}), \quad \forall\tau \ge 0, \tag{3.27}$$

where

$$\tilde{x}^e_{avg} = \frac{\varepsilon_x x^*}{\alpha c_x q_x - \varepsilon_x}, \qquad \tilde{y}^e_{avg} = \frac{\varepsilon_y y^*}{\alpha c_y q_y - \varepsilon_y}, \quad \textit{and}$$

$$e^e_{avg} = -\left[\frac{\alpha^2}{2}(q_x + q_y) + q_x\left(\frac{\varepsilon_x x^*}{\alpha c_x q_x - \varepsilon_x}\right)^2 + q_y\left(\frac{\varepsilon_y y^*}{\alpha c_y q_y - \varepsilon_y}\right)^2\right].$$

Since

$$\begin{aligned} x - x^* &= \tilde{x} + \alpha \sin(\omega t) \\ &= \big(\tilde{x} - \tilde{x}^{2\pi/\omega}\big) + \left(\tilde{x}^{2\pi/\omega} - \frac{\varepsilon_x x^*}{\alpha c_x q_x - \varepsilon_x}\right) + \frac{\varepsilon_x x^*}{\alpha c_x q_x - \varepsilon_x} + \alpha \sin \tau \end{aligned} \quad (3.28)$$

the result (3.27) implies that the first term in (3.28) converges to zero, then second term is $O(1/\bar{\omega})$, the third term is $O(\bar{\varepsilon})$ and the fourth term $O(\alpha)$. Thus, we obtain

$$\limsup_{\tau \to \infty} |x - x^*| = O(\alpha + 1/\bar{\omega} + \bar{\varepsilon}).$$

Similarly, we can obtain

$$\limsup_{\tau \to \infty} |y - y^*| = O(\alpha + 1/\bar{\omega} + \bar{\varepsilon}).$$

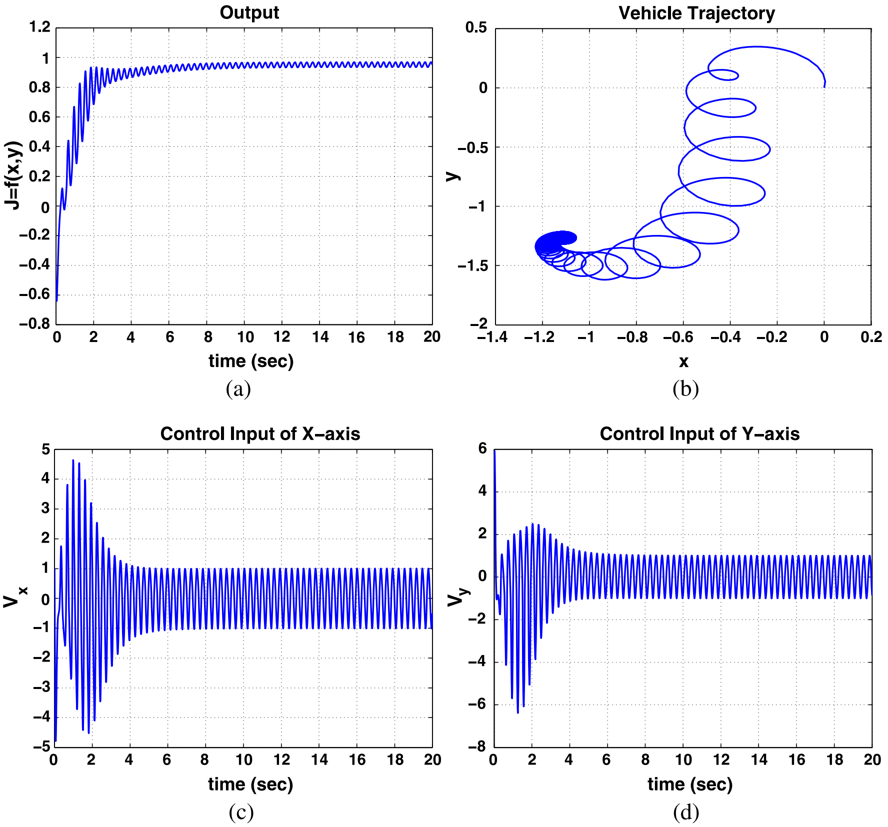
Thus, we conclude that

$$\limsup_{\tau \to \infty} |f - f^*| = O\big(\alpha^2 + (1/\bar{\omega})^2 + \bar{\varepsilon}^2\big),$$

which characterizes the asymptotic performance of the extremum seeking loop in Fig. 3.5, meaning that $(x(t), y(t))$ eventually converge to a neighborhood of the maximum. The size of the neighborhood is proportional to the amplitude of the periodic perturbation, the inverse of the perturbation frequency and the value of the unstable poles.

Example 3.1 In the simulation results shown in Fig. 3.6, we have $\varepsilon_x = \varepsilon_y = 0.05$, the perturbation frequency $\omega = 20$, perturbation amplitude $a = 0.05$, adaptation gains $c_x = c_y = 10$ and washout filter $h = 1$. The parameters of the nonlinear map (3.17) are $f^* = 1$, $q_x = 1$ and $q_y = 0.5$. The start position of the state is $(x(0), y(0)) = (0, 0)$.

The practical consequence of the perturbation signals can easily be observed in the oscillatory nature of the control signals, as seen in Figs. 3.6(c) and (d). The perturbation allows the method to locally estimate the gradient, but in practice the oscillatory controls and the resulting vehicle trajectory in Fig. 3.6(b) may not be acceptable. This is one potentially negative effect shared by all perturbation based methods, which designers should keep in mind when deciding what approach best fits the application at hand.



(a) (b)

(c) (d)

Fig. 3.6 Perturbation based extremum seeking control of a plant with slightly unstable poles: (**a**) performance output; (**b**) state; (**c**) control input of x-axis; (**d**) control input of y-axis

3.1.5 Perturbation Based Extremum Seeking Control for a Plant with Moderately Unstable Poles

Consider now the case where ε_x and ε_y in (3.16) are not restricted to be very small, but can be of "medium" size (in the sense of their relative size with respect to q_x and q_y, which may make it more difficult to satisfy the inequalities (3.26)). In this case, the robustness of the extremum seeking loop alone cannot stabilize the system, and therefore we must include a phase lead compensator to make up for the phase lag introduced by the unstable first-order dynamics. Thus, the extremum seeking scheme in Fig. 3.7 employs phase lead compensators for achieving robustness against the destabilizing effect of $\varepsilon_x, \varepsilon_y > 0$.

The transfer function of the PD compensator is designed as

$$G(s) = k_c \frac{s - z_0}{s - p_0}. \tag{3.29}$$

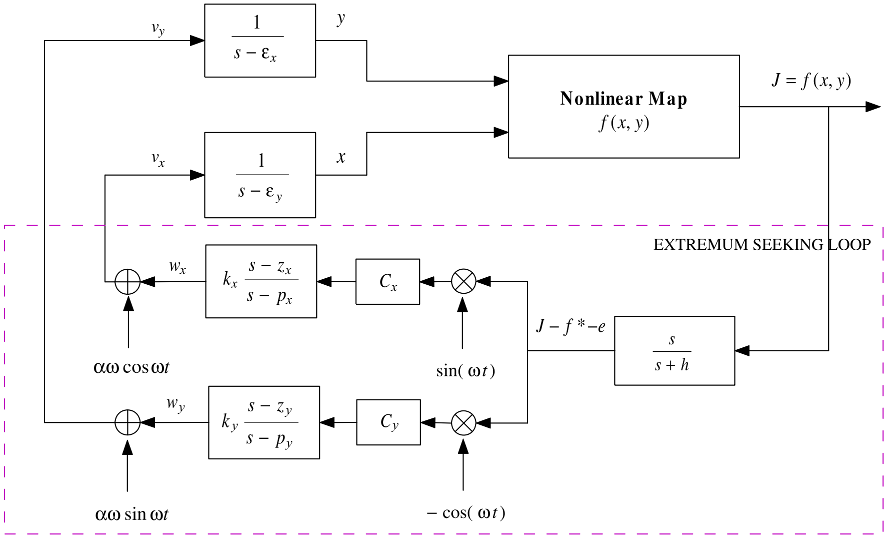

v_y
$\frac{1}{s-\varepsilon_x}$
y
Nonlinear Map
$f(x, y)$
$J = f(x, y)$
v_x
$\frac{1}{s-\varepsilon_y}$
x
EXTREMUM SEEKING LOOP
w_x
$k_x \frac{s-z_x}{s-p_x}$
C_x
$J - f^* - e$
$\frac{s}{s+h}$
$\alpha\omega \cos \omega t$
$\sin(\omega t)$
w_y
$k_y \frac{s-z_y}{s-p_y}$
C_y
$\alpha\omega \sin \omega t$
$-\cos(\omega t)$

Fig. 3.7 Perturbation based extremum seeking control of a plant with moderately unstable poles

Then, in the time scale $\tau = \omega t$, we summarize the system in Fig. 3.7 as

$$\begin{aligned}
\frac{d\tilde{x}}{d\tau} &= \frac{1}{\omega}\big[w_x + \varepsilon_x(\tilde{x} + x^* + \alpha \sin \tau)\big],\\
\frac{d\tilde{y}}{d\tau} &= \frac{1}{\omega}\big[w_y + \varepsilon_y(\tilde{y} + y^* - \alpha \cos \tau)\big],\\
\frac{de}{d\tau} &= \frac{h}{\omega}\Delta,\\
\frac{dw_x}{d\tau} &= \frac{1}{\omega}\left[p_x w_x - c_x k_x z_x \Delta \sin \tau + c_x k_x \omega \Delta \cos(\omega t) + c_x k_x \frac{d\Delta}{dt} \sin(\omega t)\right],\\
\frac{dw_y}{d\tau} &= \frac{1}{\omega}\left[p_y w_y + c_y k_y z_y \Delta \cos \tau + c_y k_y \omega \Delta \sin(\omega t) - c_y k_y \frac{d\Delta}{dt} \cos(\omega t)\right],
\end{aligned} \tag{3.30}$$

where Δ is defined in (3.22), and

$$\begin{aligned}
\frac{d\Delta}{dt} = -2q_x\big(\tilde{x} + \alpha \sin(\omega t)\big)\left(\frac{d\tilde{x}}{dt} + \alpha\omega \cos(\omega t)\right)\\
- 2q_y\big(\tilde{y} - \alpha \cos(\omega t)\big)\left(\frac{d\tilde{y}}{dt} + \alpha\omega \sin(\omega t)\right) - \frac{de}{dt}.
\end{aligned}$$

The average model of (3.30) is

$$\frac{d\tilde{x}_{avg}}{d\tau} = \frac{1}{\omega}\big[\varepsilon_x(\tilde{x}_{avg} + x^*) + w_{x\,avg}\big]$$

$$
\begin{aligned}
\frac{d\tilde{y}_{avg}}{d\tau} &= \frac{1}{\omega}\big[\varepsilon_y(\tilde{y}_{avg}+y^*)+w_{y_{avg}}\big]\\
\frac{de_{avg}}{d\tau} &= \frac{1}{\omega}(-h)\left[q_x\tilde{x}^2_{avg}+q_y\tilde{y}^2_{avg}+e_{avg}+\frac{\alpha^2}{2}(q_x+q_y)\right]\\
\frac{dw_{x\,avg}}{d\tau} &= \frac{1}{\omega}\big[(p_x-\alpha c_x k_x q_x)w_{x\,avg}+\alpha c_x k_x q_x(z_x-2\varepsilon_x+h)\tilde{x}_{avg}\\
&\quad -\alpha c_x k_x q_x\varepsilon_x x^*\big]\\
\frac{dw_{y_{avg}}}{d\tau} &= \frac{1}{\omega}\big[(p_y-\alpha c_y k_y q_y)w_{y_{avg}}+\alpha c_y k_y q_y(z_y-2\varepsilon_y+h)\tilde{y}_{avg}\\
&\quad -\alpha c_y k_y q_y\varepsilon_y y^*\big].
\end{aligned}
\tag{3.31}
$$

Then the equilibrium of the average model (3.31) is

$$
\begin{aligned}
\tilde{x}^e_{avg} &= \frac{p_x\varepsilon_x x^*}{\alpha c_x k_x q_x(z_x-\varepsilon_x+h)-p_x\varepsilon_x}\\
\tilde{y}^e_{avg} &= \frac{p_y\varepsilon_y y^*}{\alpha c_y k_y q_y(z_y-\varepsilon_y+h)-p_y\varepsilon_y}\\
e^e_{avg} &= -\frac{\alpha^2}{2}(q_x+q_y)-q_x\left(\frac{p_x\varepsilon_x x^*}{\alpha c_x k_x q_x(z_x-\varepsilon_x+h)-p_x\varepsilon_x}\right)^2\\
&\quad -q_y\left(\frac{p_y\varepsilon_y y^*}{\alpha c_y k_y q_y(z_y-\varepsilon_y+h)-p_y\varepsilon_y}\right)^2\\
w_{x\,avg}^{\;e} &= \frac{-\alpha c_x k_x q_x\varepsilon_x(z_x-\varepsilon_x+h)x^*}{\alpha c_x k_x q_x(z_x-\varepsilon_x+h)-p_x\varepsilon_x}\\
w_{y\,avg}^{\;e} &= \frac{-\alpha c_y k_y q_y\varepsilon_y(z_y-\varepsilon_y+h)y^*}{\alpha c_y k_y q_y(z_y-\varepsilon_y+h)-p_y\varepsilon_y}.
\end{aligned}
\tag{3.32}
$$

The Jacobian of (3.31) at $(\tilde{x}^e_{avg}, w_{x\,avg}^{\;e}, \tilde{y}^e_{avg}, w_{y\,avg}^{\;e}, e^e_{avg})$ is

$$
J_{avg} = \frac{1}{\omega}\begin{bmatrix}
\varepsilon_x & 1 & 0 & 0 & 0\\
a_1 & a_2 & 0 & 0 & 0\\
0 & 0 & \varepsilon_y & 1 & 0\\
0 & 0 & b_1 & b_2 & 0\\
-2hq_x x^e_{avg} & 0 & -2hq_y y^e_{avg} & 0 & -h
\end{bmatrix},
\tag{3.33}
$$

where $a_1 = \alpha c_x k_x q_x(z_x-2\varepsilon_x+h)$, $a_2 = (p_x-\alpha c_x k_x q_x)$, $b_1 = \alpha c_y k_y q_y(z_y-2\varepsilon_y+h)$ and $b_2 = (p_y-\alpha c_y k_y q_y)$. Therefore, J_{avg} will be Hurwitz if and only if the following inequalities hold:

$$\begin{aligned} \alpha c_x k_x q_x - \varepsilon_x - p_x &> 0, \\ (\alpha c_x k_x q_x + p_x)\varepsilon_x - \alpha c_x k_x q_x (z_x + h) &> 0, \\ \alpha c_y k_y q_y - \varepsilon_y - p_y &> 0, \\ (\alpha c_y k_y q_y + p_y)\varepsilon_y - \alpha c_y k_y q_y (z_y + h) &> 0, \\ h &> 0. \end{aligned} \tag{3.34}$$

If $q_x, q_y \geq \underline{q}$ and $\varepsilon_x, \varepsilon_y \leq \bar{\varepsilon}$, one possible design to satisfy the inequalities (3.34) is

1. Choose $\alpha > 0$ to be small, $h > 0$.
2. Choose $c_x > 0$, $k_x > 0$ such that $c_x k_x > \frac{\bar{\varepsilon}}{2\alpha \underline{q}}$.
3. Choose $p_x = -\alpha c_x k_x \underline{q}$ and $z_x < -h$.
4. Choose $c_y > 0$, $k_y > 0$ such that $c_y k_y > \frac{\bar{\varepsilon}}{2\alpha \underline{q}}$.
5. Choose $p_y = -\alpha c_y k_y \underline{q}$ and $z_y < -h$.

Then according to the averaging theorem [17], we have the following result.

Theorem 3.1.7 *Consider the system in Fig.* 3.7, *where the nonlinear map has the form of* (3.17). *If the conditions* (3.34) *are satisfied by design, then there exists* $\bar{\omega}$ *such that for all* $\frac{1}{\omega} \in (0, \frac{1}{\bar{\omega}})$ *the system has a unique exponentially stable periodic solution* $(\tilde{x}^{2\pi/\omega}, \tilde{y}^{2\pi/\omega}, e^{2\pi/\omega})$ *of period* $\frac{2\pi}{\omega}$ *and this solution satisfies*

$$\left\| \begin{bmatrix} \tilde{x}^{2\pi/\omega} - \tilde{x}^e_{avg} \\ \tilde{y}^{2\pi/\omega} - \tilde{y}^e_{avg} \\ e^{2\pi/\omega} - e^e_{avg} \\ {w_x}^{2\pi/\omega} - {w_x}^e_{avg} \\ {w_y}^{2\pi/\omega} - {w_y}^e_{avg} \end{bmatrix} \right\| \leq O(1/\bar{\omega}), \quad \forall \tau \geq 0, \tag{3.35}$$

where $(\tilde{x}^e_{avg}, \tilde{y}^e_{avg}, e^e_{avg}, {w_x}^e_{avg}, {w_y}^e_{avg})$ *is the equilibrium* (3.32) *of the average model* (3.31).

Since

$$\begin{aligned} x - x^* &= \tilde{x} + \alpha \sin(\omega t) \\ &= \left(\tilde{x} - \tilde{x}^{2\pi/\omega}\right) + \left(\tilde{x}^{2\pi/\omega} - \frac{p_x \varepsilon_x x^*}{\alpha c_x k_x q_x (z_x - \varepsilon_x + h) - p_x \varepsilon_x}\right) \\ &\quad + \frac{p_x \varepsilon_x x^*}{\alpha c_x k_x q_x (z_x - \varepsilon_x + h) - p_x \varepsilon_x} + \alpha \sin \tau, \end{aligned}$$

the above theorem implies that the first term converges to zero, the second term is $O(1/\bar{\omega})$, the third term is $O(\bar{\varepsilon})$ and the fourth term $O(\alpha)$, guaranteeing

$$\limsup_{\tau \to \infty} |x - x^*| = O(\alpha + 1/\bar{\omega} + \bar{\varepsilon}).$$

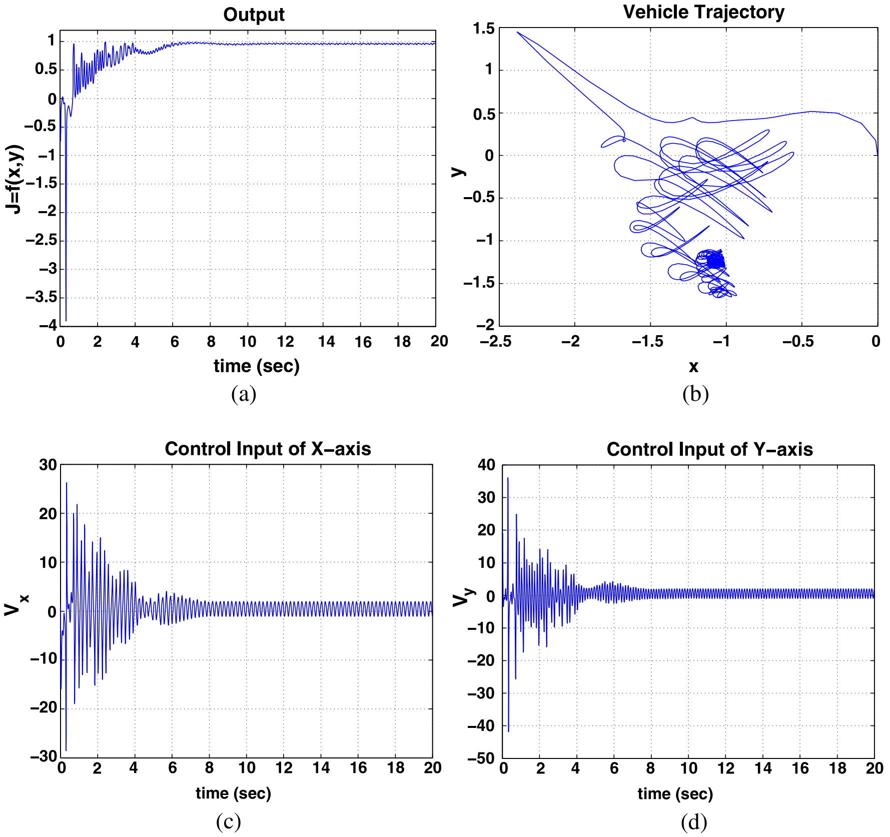


Fig. 3.8 Perturbation based extremum seeking control of a plant with moderately unstable poles: (**a**) performance output; (**b**) state; (**c**) control input of x-axis; (**d**) control input of y-axis

Similarly, we can obtain

$$\limsup_{\tau\to\infty} |y - y^*| = O(\alpha + 1/\bar{\omega} + \bar{\varepsilon}).$$

Thus, in the end we arrive at

$$\limsup_{\tau\to\infty} |f - f^*| = O\big(\alpha^2 + (1/\bar{\omega})^2 + \bar{\varepsilon}^2\big),$$

which characterizes the asymptotic performance of the extremum seeking loop in Fig. 3.7. This result implies that the system eventually converges to the neighborhood of the maximum. The size of the neighborhood is proportional to the amplitude of the periodic perturbation, the inverse of the perturbation frequency and the value of the unstable poles.

Example 3.2 The simulation results are shown in Fig. 3.8, where the two unstable poles are $\varepsilon_x = \varepsilon_y = 0.5$. The parameters of the nonlinear map (3.17) are $f^* = 1$, $q_x = 1$ and $q_y = 0.5$. The parameters of the PD compensator (3.29) are $k_c = 2$, $z_0 = -5$, $p_0 = -1$, the perturbation frequency $\omega = 30$, perturbation amplitude $a = 0.05$, adaptation gain $c_x = c_y = 15$ and washout filter $h = 1$. The system's start position is set to $(0, 0)$.

3.2 Numerical Optimization Based Extremum Seeking Control

In this section, we consider the minimum seeking control design of nonlinear system (3.1) with performance function (3.2). Here, we treat extremum seeking control from the perspective of optimization. From this point of view, extremum seeking control can be considered as a type of constrained optimization problem, whose constraint is the differential equation (3.1) as compared to the traditional algebraic constraints, and the manipulation of x has to be done indirectly through the control input u. The extremum seeking control problem then can be stated as

$$\min_{x \in \mathbb{R}^n} \quad J(x) \quad \text{subject to} \quad \dot{x} = f(x, u).$$

Now the state x is feasible only if it is a solution of the dynamic system. In the case when (3.1) is controllable, there always exists an input u that transfers x to anywhere in $\mathbb{R}^n$ in a finite time. Although controllable dynamic system constraints do allow x to be anywhere in the state space where the numerical optimizer wants, the way in which x reaches the particular place is determined by the dynamic system and the state regulator to be designed. Therefore, the goal of extremum seeking control is to design a controller based on output measurements and state measurements to regulate the state to an unknown minimizer of an unknown performance function.

Assumption 3.2.1 (Existence of the Minimum) *The performance function $J(x)$ is continuous on the compact level sets $L(x_0^s) = \{x \in \mathbb{R}^n | J(x) \leq J(x_0^s)\}$ for all x_0^s in $\mathbb{R}^n$.*

Assumption 3.2.2 (Isolated Minimizer) *The global minimizer $x^* \in \mathbb{R}^n$ of $J(x)$ is an isolated minimizer.*

Assumption 3.2.3 (Stabilizable Equilibrium) *The global minimizer $x^* \in \mathbb{R}^n$ of $J(x)$ is a stabilizable equilibrium point of the closed-loop system.*

Assumption 3.2.1 guarantees the existence of the minimum, and any numerical optimization algorithms with first-order global convergence property will produce a sequence $\{x_k^s\}$ converging to a minimizer (more precisely, a first-order stationary point) of the performance function. Assumptions 3.2.2 and 3.2.3 are required for the applicability of Lyapunov analysis. Assumption 3.2.3 also ensures that there is a

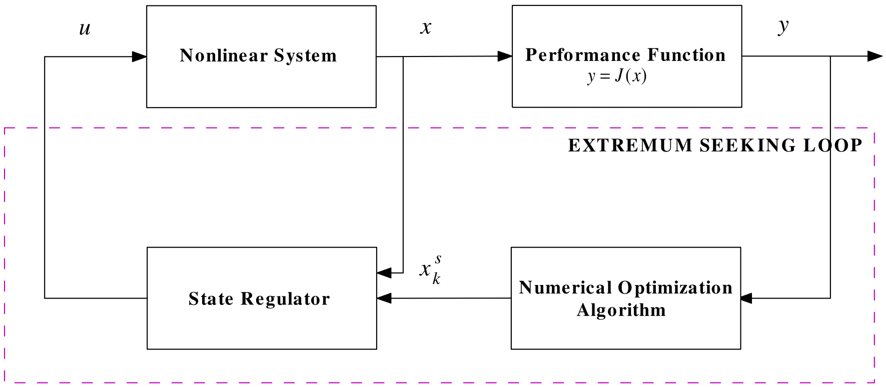


Fig. 3.9 Numerical optimization based extremum seeking control

controller that can operate the system at the desired set point x^*; otherwise, we may still be able to design a controller that enables the system to track a reference trajectory containing the minimizer x^*. In the latter case, the steady state performance output will be oscillating. Note that Assumption 3.2.3 is more practical than the assumptions in Chap. 5 of [2] (presented in Assumptions 3.1.1 and 3.1.2 in Sect. 3.1), where the closed-loop system is assumed to have an equilibrium set parameterized by the control argument and extremum seeking is achieved by tuning the single parameter.

A block diagram of numerical optimization based extremum seeking control is proposed in Fig. 3.9, where the nonlinear system is modeled as (3.1) and the performance function is (3.2). The extremum seeking controller (state regulator) ensures that the state x travels along the set point sequence $\{x_k^s\}$ generated by the numerical optimization algorithm, which eventually converges to a minimizer of the performance function. A basic framework for such extremum seeking control is as follows.

3.2.1 Basic Numerical Optimization based Extremum Seeking Control Algorithm

Step 0 Given x_0^s, set $t_0 = 0$, $x(t_0) = x_0^s$ and $k = 0$.

Step 1 Use an optimization algorithm to produce x_{k+1}^s based on current state $x(t_k)$, the measurements of $J(x(t_k))$ and/or $\nabla J(x(t_k))$. Denote

$$x_{k+1}^s = \text{OPTIMIZER}\big(x(t_k)\big).$$

Step 2 Design a state regulator $u(t)$ that regulates the state $x(t_k)$ to x_{k+1}^s in a finite time δ_k, let $t_{k+1} = t_k + \delta_k$. That is, $x(t_{k+1}) = x_{k+1}^s$.

Step 3 Set $k \leftarrow k+1$. Go to Step 1 until convergence.

Here we have only introduced the basic framework for NOESC. This framework will be thoroughly explained and expanded upon in Chaps. 4 and 5, where the class of systems we consider is explained, and the contents of the state regulator block are addressed. Moreover, in the next two chapters we also deal with the important issues of convergence and robustness of the NOESC algorithms.

References

1. Adetola, V., Dehaan, D., Guay, M.: Adaptive extremum seeking receding horizon control of nonlinear system. In: Proceedings of the American Control Conference, pp. 2937–2942 (2004)
2. Ariyur, K.B., Krstić, M.: Real-time Optimization by Extremum-seeking Control. Wiley-Interscience, Hoboken (2003)
3. Banavar, R.N.: Extremum seeking loops with assumed functions: estimation and control. In: Proceedings of the American Control Conference, vol. 4, pp. 3159–3164 (2002)
4. Banavar, R.N., Chichka, D.F., Speyer, J.L.: Convergence and synthesis issues in extremum seeking control. In: Proceedings of the American Control Conference, vol. 1, pp. 438–443 (2000)
5. Banavar, R.N., Chichka, D.F., Speyer, J.L.: Functional feedback in an extremum seeking loop. In: Proceedings of the 40th Conference on Decision and Control, vol. 2, pp. 1316–1321 (2001)
6. Banavar, R.N., Chichka, D.F., Speyer, J.L.: Convergence and synthesis issues in extremum seeking control. Int. J. Adapt. Control Signal Process. **17**, 751–762 (2003)
7. Cortés, J.: Achieving coordination tasks in finite time via nonsmooth gradient flows. In: Proceedings of Conference on Decision and Control (2005)
8. Drakunov, S.V., Özgüner, Ü.: Optimization of nonlinear system output via sliding mode approach. In: Proceedings of the IEEE International Workshop on Variable Structure and Lyapunov Control of Uncertain Dynamical Systems, Sheffield, UK, pp. 61–62 (1992)
9. Drakunov, S.V., Özgüner, Ü., Dix, P., Ashrafi, B.: ABS control using optimum search via sliding modes. IEEE Trans. Control Syst. Technol. **3**, 79–85 (1995)
10. Chin, Y.K. et al.: Sliding mode ABS wheel slip control. In: Proceedings of the American Control Conference, pp. 1–6 (1992)
11. Guay, M., Zhang, T.: Adaptive extremum seeking control of nonlinear dynamic systems with parametric uncertainties. Automatica **39**(7), 1283–1293 (2003)
12. Haskara, I., Hatipoglu, C., Özgüner, Ü.: Sliding mode compensation, estimation and optimization methods in automotive control. In: Yu, X., Xu, J.-X. (eds.) Variable Structure Systems: Towards the 21st Century. Lecture Notes in Control and Information Sciences, vol. 274, pp. 155–174 (2002)
13. Haskara, I., Özgüner, Ü., Winkelman, J.: Extremum control for optimal operating point determination and set point optimizing via sliding modes. J. Dyn. Syst. Meas. Control **122**(4), 719–724 (2000)
14. Haskara, I., Özgüner, Ü., Winkelman, J.: Wheel slip control for antispin acceleration via dynamic spark advance. Control Eng. Pract. **8**(10), 1135–1148 (2000)
15. Hirsch, M.W., Smale, S., Devaney, R.L.: Differential Equations, Dynamical Systems and an Introduction to Chaos. Academic Press, New York (2004)
16. Hirsch, W.M., Smale, S.: Differential Equations, Dynamical Systems and Linear Algebra. Academic Press, New York (1974)
17. Khalil, H.K.: Nonlinear Systems. Prentice Hall, Upper Saddle River (2001)
18. Korovin, S.K., Utkin, V.I.: Using sliding modes in static optimization and nonlinear programming. Automatica **10**, 525–532 (1974)
19. Krstić, M., Wang, H.-H.: Design and stability analysis of extremum seeking feedback for general nonlinear systems. Automatica **36**(2), 595–601 (2000)

20. Pan, Y., Özgüner, Ü.: Discrete-time extremum seeking algorithms. In: Proceedings of the American Control Conference, vol. 4, pp. 3147–3152 (2002)
21. Pan, Y., Özgüner, Ü.: Sliding mode extremum seeking control for linear quadratic dynamic game. In: Proceedings of the American Control Conference, pp. 614–619 (2004)
22. Pan, Y., Acarman, T., Özgüner, Ü.: Nash solution by extremum seeking control approach. In: Proceedings of the Conference on Decision and Control, vol. 1, pp. 329–334 (2002)
23. Pan, Y., Özgüner, Ü., Acarman, T.: Stability and performance improvement of extremum seeking control with sliding mode. Int. J. Control **76**(9/10), 968–985 (2003)
24. Speyer, J.L., Banavar, R.N., Chichka, D.F., Rhee, I.: Extremum seeking loops with assumed functions. In: Proceedings of the 39th Conference on Decision and Control, vol. 1, pp. 142–147 (2000)
25. Tan, H.S., Tomizuka, M.: An adaptive sliding mode vehicle traction controller design. In: Proceedings of the American Control Conference, pp. 1035–1058 (1989)
26. Utkin, V.I.: Sliding Modes in Control and Optimization. Springer, Berlin (1994)
27. Winkelman, J., Haskara, I., Özgüner, Ü.: Tuning for dynamic spark advance control. In: Proceedings of the American Control Conference, pp. 163–164 (1999)
28. Yamanka, S., Ohmori, H.: Nonlinear adaptive extremum seeking control for time delayed index in the presence of deterministic disturbance. In: Proceedings of the PhysCon 2005, pp. 121–125 (2005)
29. Yu, H., Özgüner, Ü.: Extremum-seeking control strategy for ABS system with time delay. In: Proceedings of the American Control Conference, vol. 5, pp. 3753–3758 (2002)
30. Yu, H., Özgüner, Ü.: Extremum-seeking control via sliding mode with periodic search signals. In: Proceedings of the Conference on Decision and Control, vol. 1, pp. 323–328 (2002)
31. Yu, H., Özgüner, Ü.: Smooth extremum-seeking control via second order sliding mode. In: Proceedings of the American Control Conference, vol. 4, pp. 3248–3253 (2003)
32. Zhang, C., Siranosian, A., Krstić, M.: Extremum seeking for moderately unstable systems and for autonomous target tracking without position measurements. Automatica **43**, 1832–1839 (2007)

Chapter 4
Finite Time State Regulator Design

4.1 Finite Time State Regulator Design

4.1.1 Linear Time Invariant Systems

Here, we consider a single-input, single-output (SISO) linear time invariant (LTI) system of the general form

$$\dot{x} = Ax + Bu \tag{4.1}$$

with the performance function

$$y = J(x), \tag{4.2}$$

as it was defined in (3.2) and repeated here for convenience, where $x \in \mathbb{R}^n$ is the state, and $u \in \mathbb{R}$ is the input. The matrices A, B are given as a model of a real system. However, the explicit form of the performance function $J(x)$ and its minimum are not known, and we assume we are only able to measure the function value y or its derivatives. We need the following assumption to ensure the feasibility of extremum seeking control for the LTI system (4.1).

Assumption 4.1.1 *The LTI system* (4.1) *is controllable.*

Now, we can combine an optimization algorithm and a state regulator originated from the controllability theorem in [1] to form an extremum seeking control scheme:

4.1.1.1 NOESC Scheme for LTI Systems

Step 0 Given x_0^s, choose a termination threshold ε_0, and let $t_0 = 0$, $x(t_0) = x_0^s$, and set the index $k = 0$.

Step 1 Use an optimization algorithm with first-order global convergence to produce x_{k+1}^s based on current state $x(t_k)$, the measurement of $J(x(t_k))$ or

C. Zhang, R. Ordóñez, *Extremum-Seeking Control and Applications*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2224-1_4, © Springer-Verlag London Limited 2012

$\nabla J(x(t_k))$. Denote

$$x_{k+1}^s = \text{OPTIMIZER}\bigl(x(t_k)\bigr).$$

Step 2 Choose a regulation time δ_k, let $t_{k+1} = t_k + \delta_k$, and design the control input during $t_k \le t \le t_{k+1}$ to be

$$u(t) = -B^\top e^{A^\top (t_{k+1}-t)} W_c^{-1}(\delta_k)\bigl[e^{A\delta_k} x(t_k) - x_{k+1}^s\bigr], \tag{4.3}$$

where the controllability Gramian is given by

$$W_c(\delta_k) = \int_0^{\delta_k} e^{A\tau} B B^\top e^{A^\top \tau}\, d\tau. \tag{4.4}$$

Step 3 If $\|\nabla J(x(t_{k+1}))\| < \varepsilon_0$, then stop. Otherwise, set $k \leftarrow k+1$. Go to Step 1.

Remark 4.1.2 The above extremum seeking control scheme can be derivative-free if the optimization algorithm used in Step 1 (as well as the termination criterion in Step 3) does not require gradient information. For example, we can use derivative-free trust region methods [2, 7] or direct search [5]. Some modifications of the above scheme are required in order to use the trust region methods due to the need to obtain the ratio ρ_k in (2.9), where we may need additional regulation time to drive the state back to x_k^s if $\rho_k \le \eta$. The reader should refer to Sect. 4.2.2 and [9, 10] for the details of trust region based extremum seeking control.

Remark 4.1.3 If the steepest descent method is used in Step 1, we will have

$$x_{k+1}^s = \text{OPTIMIZER}\bigl(x(t_k)\bigr) = x(t_k) - \alpha_k \nabla J\bigl(x(t_k)\bigr).$$

Even though it requires gradient measurement $\nabla J(x(t_k))$ by using steepest descent method, such measurement is only needed every δ_k time, therefore, we can estimate the gradient by collecting enough measurements of J during the δ_k time.

Remark 4.1.4 The stopping criterion $\|\nabla J(x(t_{k+1}))\| < \varepsilon_0$ is used only for simplicity, where ε_0 is a predefined small positive constant. In case gradient information is not available, there are other stopping criteria only based on the difference of function values [6].

Now, we present the convergence analysis of the NOESC scheme.

Theorem 4.1.5 *Consider the LTI system* (4.1) *with performance output* (4.2). *Assume the LTI system* (4.1) *satisfies* 4.1.1 *and the performance function* (4.2) *satisfies Assumption* 3.2.1. *If the extremum seeking control scheme above is applied, where the optimization algorithm used is of first-order global convergence, then the state x will globally asymptotically converge to the first-order stationary point of the performance function* (4.2).

Proof First, given that the performance function (4.2) satisfies Assumption 3.2.1, an optimization algorithm with first-order global convergence will produce a search sequence $\{x_k^s\}$ that globally asymptotically converges to the first-order stationary point of the performance function (4.2).

By assuming the LTI system (4.1) is controllable, the controller (4.3) is feasible since the controllability Gramian $W_c(\delta_k)$ in (4.4) is nonsingular. First, at $t = t_0$, the state $x(t_0) = x_0^s$, we have $x_1^s = \text{OPTIMIZER}(x(t_0))$. Then, at time $t_1 = t_0 + \delta_0$, we will have

$$x(t_1) = e^{A\delta_0}x(t_0) + \int_{t_0}^{t_1} e^{A(t_1-\tau)}Bu(\tau)\,d\tau = x_1^s,$$

where the control input during (t_0, t_1) is

$$u(t) = -B^\top e^{A^\top(t_1-t)}\left[\int_0^{\delta_0} e^{A\tau}BB^\top e^{A^\top\tau}\,d\tau\right]^{-1}\left[e^{A\delta_0}x(t_0) - x_1^s\right].$$

By induction, at $t = t_k$, we suppose the state $x(t_k) = x_k^s$, then we obtain $x_{k+1}^s = \text{OPTIMIZER}(x(t_k))$. At time $t_{k+1} = t_k + \delta_k$, we will have

$$x(t_{k+1}) = e^{A\delta_k}x(t_k) + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}Bu(\tau)\,d\tau = x_{k+1}^s,$$

where $u(t)$ is defined in (4.3). Thus the controller (4.3) interpolates between each point in the sequence $\{x_k^s\}$ precisely, within each time interval of length δ_k. Therefore, the state of the system will globally asymptotically converge to the first-order stationary point of performance function (4.2). □

Remark 4.1.6 The convergence result for the extremum seeking control scheme is global since the numerical optimization algorithm used is of first-order global convergence.

Remark 4.1.7 Additional assumptions about the performance function J may be required to guarantee that an arbitrary optimization algorithm with first-order global convergence indeed converges to the stationary point of (4.2). For example, according to Theorem 2.2.6, we need to assume $J(x)$ is continuously differentiable and $\nabla J(x)$ is Lipschitz continuous for line search methods; also we will assume J is Lipschitz, continuously differentiable and bounded from below on level sets when using trust region methods.

Remark 4.1.8 The design of controller (4.3) is not limited to single-input systems, and it is just one way to fulfill the state regulation task. It is an open-loop controller during the time interval of length δ_k, and it does not consider the change of x_{k+1}^s, which is fed back from the optimization algorithm. This approach has the advantage of achieving regulation in a finite time, but it relies on the precise knowledge of the A and B matrices, and is very difficult to use it to yield a robust design because of its open-loop nature. Later on, we will relax the state regulation design criterion from perfect regulation to regulation within a neighborhood of x_{k+1}^s, which provides further flexibility of using other designs of state regulator to deal with input disturbance or unmodeled plant dynamics.

Remark 4.1.9 The only requirement for the feasibility of state regulator (4.3) is that the LTI system is controllable (in other words, that the controllability Gramian $W_c(\delta_k)$ of (4.4) is nonsingular), which means the extremum seeking control scheme works for both stable and unstable systems. However, in practice it is preferable to first stabilize the unstable system by pole placement, then design a state regulator on the stabilized LTI system. The reason is that the unstable LTI system will amplify the regulation error resulting from an input disturbance, for example.

Remark 4.1.10 The performance of the extremum seeking control design largely depends on the performance function to be optimized, the optimization algorithm used, and a robust and efficient state regulator.

Remark 4.1.11 If the performance function (4.2) is differentiable and convex, then the convergence to stationary point becomes convergence to the global minimum [6].

Example 4.1 Now, consider a second order stable LTI system in its controllable canonical form. Let $x = [x_1, x_2]^\top$, and

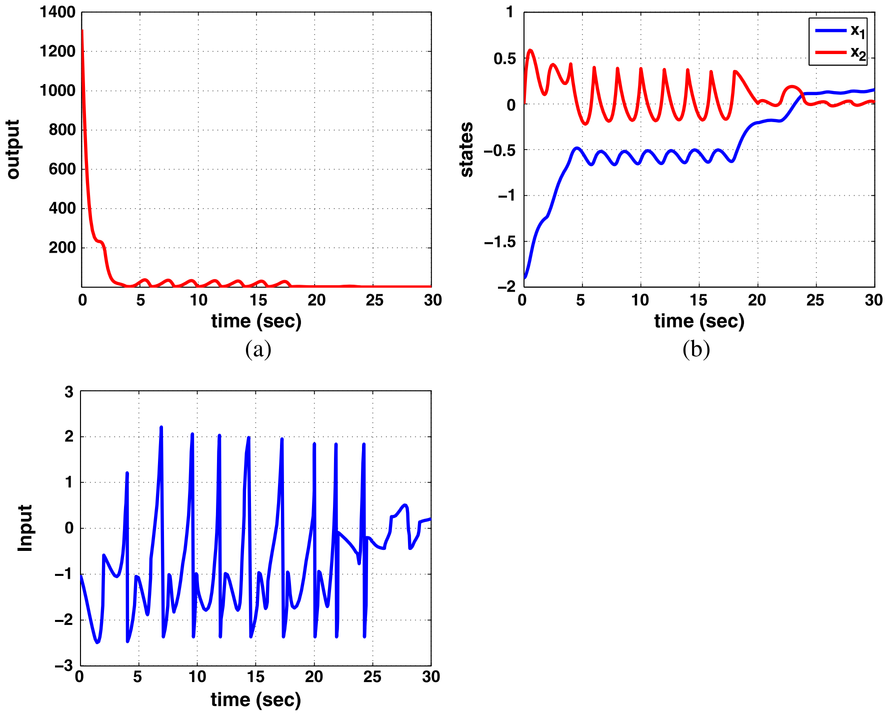
$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \tag{4.5}$$

$$y = J(x) = 100\big(x_2 - x_1^2\big)^2 + (1 - x_1)^2. \tag{4.6}$$

The banana function (4.6) has its minimizer at $x^* = [1, 1]^\top$, and $J(1, 1) = 0$. The explicit form of the function $J(x)$ and its minimum are both unknown to the designer. Extremum seeking control scheme based on steepest descent algorithm with inexact line search is applied. The simulation results with the first 15 steps are shown in Fig. 4.1, where $x_0^s = [-1.9, 0]^\top$, and $\delta_k = 2$. Hence, we only require the gradient measurement at $x(t_k)$ every δ_k time to implement the optimization algorithm.

The performance output (Fig. 4.1(a)) approaches its minimum at $J(1, 1) = 0$ and the state (Fig. 4.1(b)) accordingly converges to the minimizer $[1, 1]^\top$. The steepest descent algorithm produces a sequence $\{x_k^s\}$ of set-point commands for the controller to follow. The trajectory between x_k^s and x_{k+1}^s is shaped by the dynamical system (4.1) and the regulator (4.3). This can be clearly viewed in Fig. 4.2, where the blue circle represents the command sequence $\{x_k^s\}$ and the red dashed line represents the state trajectory.[1] The choice of δ_k is rather heuristic in this example. However, in practice this is an important design factor. We can see that a smaller

[1] Since it would take thousands of steps for the steepest descent algorithm to converge to the minimizer of the banana function, we only simulate the first 15 steps for illustrative purposes.



(a)

(b)

(c)

Fig. 4.1 Extremum seeking control for a LTI system: (**a**) performance output; (**b**) states; (**c**) control input

δ_k yields a larger control force to satisfy the short regulation time. Thus, δ_k should be chosen such that the control force does not exceed the practical limits. There is always a tradeoff between the extremum seeking time and the control gain.

Example 4.2 As another example, consider again the second order linear system (4.5), although this time with the cost function

$$y = J(x) = 5x_1^2 + x_2^2 + 4x_1x_2 - 14x_1 - 6x_2 + 20. \tag{4.7}$$

The performance function $J(x)$ has its minimizer at $x^* = (1,1)$ and $J(1,1) = 10$. Here we use, for simplicity, a steepest descent algorithm with exact line search [3]. The search direction can be computed to be

$$p_k = -\nabla J\left(x_1^k, x_2^k\right) = \left[-10x_1^k - 4x_2^k + 14, -2x_2^k - 4x_1^k + 6\right]^\top = \left[p_1^k, p_2^k\right]^\top.$$

Then, we can derive an explicit expression of the step length

$$\alpha_k = \text{argmin}_\alpha\, f(x_k + \alpha p_k) = \frac{(p_1^k)^2 + (p_2^k)^2}{2(5(p_1^k)^2 + (p_2^k)^2 + 4p_1^k p_2^k)}.$$

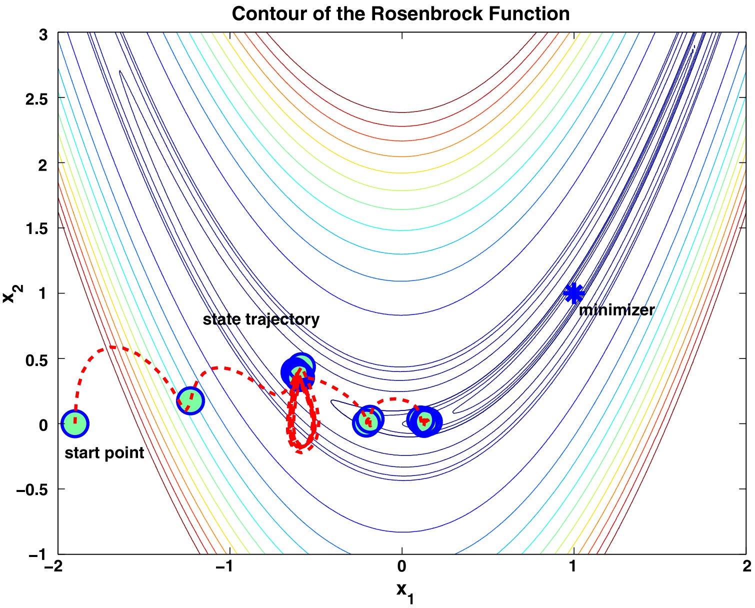


Fig. 4.2 Extremum seeking control for a LTI system: phase portrait of the steepest descent sequence $\{x_k^s\}$ over the contour of the performance function

Note that, generally speaking, exact line search is not possible since the cost function is assumed unknown; however, as shown in Example 4.1, similar results are expected for inexact line search as well as other optimization algorithms.

Given $\delta_k = 2$, $x_0 = [-10, 10]$, the simulation results are shown in Fig. 4.3. The performance function (Fig. 4.3(a)) approaches its minimum at $J(1, 1) = 10$. The steepest descent algorithm produces a sequence $\{x_k\}$ as a set-point sequence for the controller to follow. The trajectory between x_k and x_{k+1} is shaped by the dynamical system constraints. This can be clearly viewed in Fig. 4.4, where the blue circle represents the sequence $\{x_k\}$ and the red dashed line represents the state trajectory.

A point of interest here is the oscillatory behavior observed in Fig. 4.3(b), where the states are plotted. The reason for this oscillation is that the minimizer $x^* = (1, 1)$ of (4.7) is not an equilibrium point of system (4.5). Thus, the best the controller can do is force the state to periodically revisit the minimizer.

4.1.2 State Feedback Linearizable Systems

Now, we consider a SISO nonlinear affine system

$$\dot{x} = f(x) + g(x)u \tag{4.8}$$

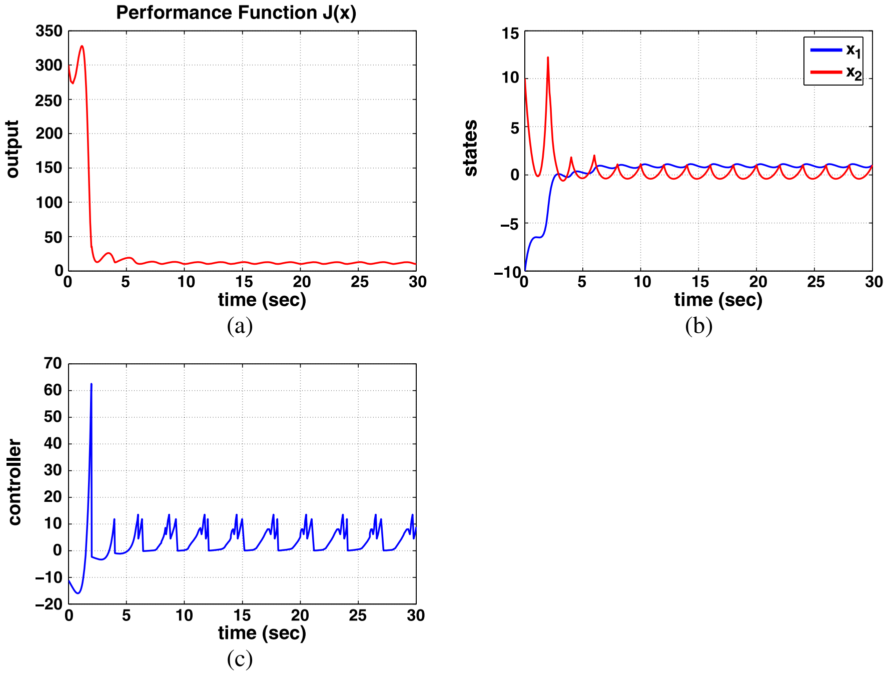


Fig. 4.3 Extremum seeking control for LTI system: (**a**) performance function; (**b**) states; (**c**) control input

with the performance function defined in (4.2), where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}$ is the input, $f, g : D \to \mathbb{R}^n$ are smooth functions on a domain D. We have the following assumption for the nonlinear affine system:

Assumption 4.1.12 *The nonlinear affine system is state feedback linearizable on the domain D.*

From this assumption it follows that we can always put the system in controllable canonical form [4]. That is, there exists a diffeomorphism $T : D \to \mathbb{R}^n$ such that $D_z = T(D)$ contains the origin and the change of variables $z = T(x)$ transforms the system (4.8) into the form

$$\dot{z} = Az + B\gamma(x)\big[u - \alpha(x)\big] \tag{4.9}$$

with (A, B) in controllable canonical form, that is,

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \tag{4.10}$$

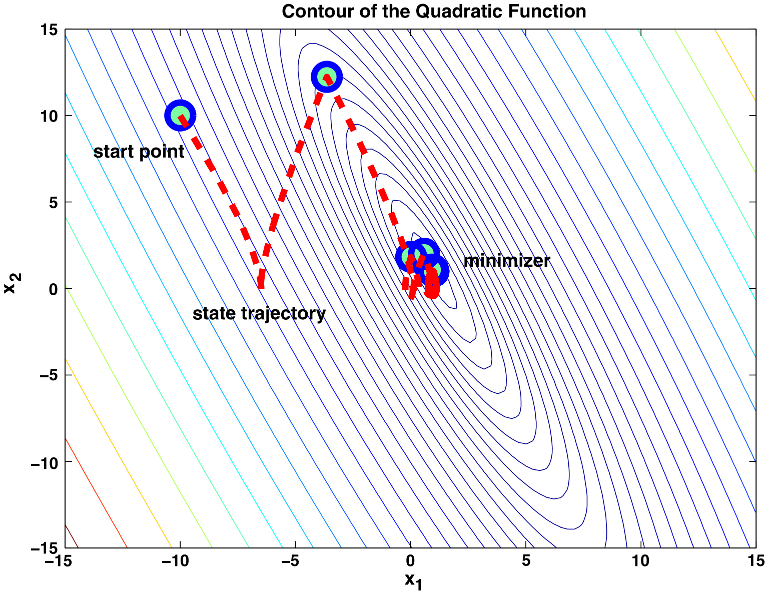


Fig. 4.4 Extremum seeking control for LTI system: phase portrait of the steepest descent sequence $\{x_k\}$ over the contour of the performance function

and $\gamma(x)$ nonsingular for all $x \in D$. Then we can easily extend the results on LTI systems to outline the extremum seeking control scheme for the state feedback linearizable systems (4.8), where we still assume $D = \mathbb{R}^n$ for simplicity:

4.1.2.1 NOESC Scheme for State Feedback Linearizable Systems

Step 0 Given x_0^s, set ε_0, $t_0 = 0$, $x(t_0) = x_0^s$ and $k = 0$.
Step 1 Let $x_{k+1}^s = \text{OPTIMIZER}(x(t_k))$.
Step 2 Set $z_{k+1}^s = T(x_{k+1}^s)$.
Step 3 Choose a regulation time δ_k, let $t_{k+1} = t_k + \delta_k$, and choose the control input during $t_k \le t \le t_{k+1}$ to be

$$u(t) = \alpha(x) + \gamma^{-1}(x)v(t), \tag{4.11}$$

where

$$v(t) = -B^\top e^{A^\top(t_{k+1}-t)} W_c^{-1}(\delta_k)\left[e^{A\delta_k} T\big(x(t_k)\big) - z_{k+1}^s\right], \tag{4.12}$$

and $W_c(\delta_k) = \int_0^{\delta_k} e^{A\tau} B B^\top e^{A^\top \tau}\, d\tau$.
Step 4 If $\|\nabla J(x(t_{k+1}))\| < \varepsilon_0$, then stop. Otherwise, set $k \leftarrow k+1$. Go to Step 1.

Theorem 4.1.13 *Consider the nonlinear affine system* (4.8) *with performance output* (4.2). *Suppose the system* (4.8) *satisfies Assumption* 4.1.12 *and the performance* *function* (4.2) *satisfies Assumption* 3.2.1; *moreover, if the extremum seeking control scheme above is applied*, *where the optimization algorithm used is of first-order global convergence*, *then the state x will globally asymptotically converge to the first-order stationary point of the performance function* (4.2).

The proof mainly follows the proof of Theorem 4.1.5. The feasibility of the controller defined in (4.11) is guaranteed by Assumption 4.1.12. Remarks similar to those of 4.1.2 through 4.1.11 also apply here.

4.1.3 Input–Output Feedback Linearizable Systems

We can extend the previous results on state feedback linearizable systems to input–output feedback linearizable systems given some minor modifications. We make the following assumption:

Assumption 4.1.14 *The nonlinear affine system* (4.8) *is input–output feedback linearizable from input u to output $\bar{y}$ on D.*

We define

$$\bar{y} = h(x), \tag{4.13}$$

where $h : D \to \mathbb{R}$ is sufficiently smooth in the domain D. The motivation of defining a new output $\bar{y}$ is to retain the claim that in extremum seeking control we do not necessarily have knowledge of the performance function, but we do need the knowledge of some suitable output $\bar{y} = h(x)$ to perform input–output linearization. Let d be the relative degree of the nonlinear system (4.13). Then, for every $x_0^s \in D$, a neighborhood N of x_0^s and a diffeomorphism $T : N \to \mathbb{R}^n$ exist such that the change of variables $z = [\eta, \xi] = T(x)$, transforms the system (4.13) into the form

$$\dot{\eta} = f_0(\eta, \xi), \tag{4.14}$$

$$\dot{\xi} = A\xi + B\gamma(x)\big[u - \alpha(x)\big], \tag{4.15}$$

$$\bar{y} = C\xi, \tag{4.16}$$

where $\xi \in \mathbb{R}^d$, $\eta \in \mathbb{R}^{n-d}$, the pair (A, B) is in controllable canonical form (4.10), and $\gamma(x)$ is nonsingular in N. Since η is uncontrollable, in order to fulfill the extremum seeking of the performance function, two more assumptions are proposed:

Assumption 4.1.15 *The performance function is not dependent on the state η of the internal dynamics.*

Assumption 4.1.16 *The state η of the internal dynamics* (4.14) *will be bounded given bounded ξ and any initial state x_0^s.*

The Assumption 4.1.15 puts the performance function in the form

$$y = J(x) = J_d(\xi).$$

It is a reasonable assumption since we have to be able to control the variables of the performance function to achieve the extremum seeking. Moreover, Assumption 4.1.16 simply means that the internal dynamics are well behaved such that the linearizing control signal will not grow unbounded due to the existence of the uncontrollable state η. Note that assuming input-to-state stability [4] of the internal dynamics is stronger than Assumption 4.1.16. However, simply assuming the zero dynamics $\dot{\eta} = f_0(\eta, 0)$ are asymptotically stable is also not enough since the state η may grow unbounded given bounded input ξ. Thus, the same analysis for the state feedback linearizable systems holds here given the extremum seeking scheme in Sect. 4.1.2 with minor modifications, where we replace z with ξ. The following theorem is a straightforward extension of Theorem 4.1.13.

Theorem 4.1.17 *Consider the nonlinear affine system* (4.8) *with performance output* (4.2). *Suppose the system* (4.8) *satisfies Assumptions* 4.1.14 *and* 4.1.16, *and the performance function* (4.2) *satisfies Assumptions* 3.2.1 *and* 4.1.15; *moreover, if the extremum seeking control scheme of Sect.* 4.1.2 *is applied with z replaced by ξ, then the state ξ will globally asymptotically converge to the first-order stationary point of the performance function* (4.2).

The proof mainly follows the proof of Theorem 4.1.5. The feasibility of the controller defined in (4.11) is guaranteed by Assumptions 4.1.14 and 4.1.16. Similar remarks like 4.1.2 through 4.1.11 also apply here.

4.2 Robustness Issues

The main restriction of Theorems 4.1.5, 4.1.13 and 4.1.17 is the requirement of perfect state regulation to guarantee convergence. That is, at each iteration, the controller needs to regulate the state precisely to the desired set-point x^s_{k+1}, which is produced by the iterative optimization algorithm based on the current state $x(t_k)$. In practical applications, noisy output or state measurements, input disturbances, saturation and time delay, unmodeled plant dynamics and computational errors will be detrimental to the theoretical result. Thus, we can only expect to be able to regulate the state to a neighborhood of the set-point x^s_{k+1}.

For example, let us consider a LTI system with input disturbance. Let $\hat{u}(t) = u(t) + \Delta u(t)$, where $u(t)$ is given as in (4.3). At time $t = t_0$, the state $x(t_0) = x^s_0$ and $x^s_1 = \text{OPTIMIZER}(x(t_0))$. Then, at time $t_1 = t_0 + \delta_0$, we will have

$$x(t_1) = e^{A\delta_0}x(t_0) + \int_{t_0}^{t_1} e^{A(t_1-\tau)}B\hat{u}(\tau)\,d\tau$$

$$
\begin{aligned}
&= e^{A\delta_0}x(t_0) + \int_{t_0}^{t_1} e^{A(t_1-\tau)}Bu(\tau)\,d\tau + \int_{t_0}^{t_1} e^{A(t_1-\tau)}B\Delta u(\tau)\,d\tau \\
&= x_1^s + e_1,
\end{aligned}
$$

where $e_1 = \int_{t_0}^{t_1} e^{A(t_1-\tau)}B\Delta u(\tau)\,d\tau$. However, the error will not accumulate since the optimization algorithm will generate the next destination based on the current state $x(t_1)$. That is, by including the numerical optimizer in the extremum seeking loop, it offers a feedback mechanism to robustify the extremum seeking scheme. Denote $\hat{x}_0^s = x(t_0) = x_0^s$, $\hat{x}_1^s = x(t_1) = x_1^s + e_1$. Then, by induction, we find that the controller interpolates precisely between the sequence $\{\hat{x}_k^s\}$, where

$$
\begin{aligned}
\hat{x}_{k+1}^s &= x(t_{k+1}) = x_{k+1}^s + e_{k+1}, \\
x_{k+1}^s &= \text{OPTIMIZER}\big(\hat{x}_k^s\big), \\
e_{k+1} &= \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}B\Delta u(\tau)\,d\tau.
\end{aligned}
$$

The error e_{k+1} will be bounded if $\delta_k = t_{k+1} - t_k$ is bounded and the input disturbance is bounded. Moreover, for stable systems, the system matrix A has eigenvalues with negative real part, and therefore the state transition matrix has exponentials terms that decrease with time. Therefore, e_{k+1} will asymptotically converge to some constant as t_{k+1} approaches infinity, that is, even if δ_k is unbounded, we will still have a bounded e_{k+1} given bounded input disturbance. Moreover, the more negative the real parts of the eigenvalues are, the smaller e_{k+1} will be.

On the other hand, for unstable systems, the state transition matrix will amplify even a small input disturbance Δu, and e_{k+1} will grow as δ_k increases. Therefore, we wish to have a stable LTI system and a short regulation time. Consequently, there is the need of a high gain controller to deal with the disturbance. Consider, for instance, extremum seeking control of an unstable but controllable LTI system. In this case, we would perform pole placement to transform the unstable LTI system to a desired stable LTI system, and then design the state regulator on the stabilized LTI system.

Similarly, for state feedback linearizable systems, given the knowledge of A and B, which are in controllable canonical form, the controller will be implementing functions $\hat{\alpha}(x)$ and $\hat{\gamma}(x)$, approximations of $\alpha(x)$ and $\gamma(x)$. That is,

$$
u(t) = \hat{\alpha}(x) + \hat{\gamma}^{-1}(x)v(t),
$$

where $v(t)$ is defined in (4.12). Now the closed-loop systems becomes

$$
\begin{aligned}
\dot{z} &= Az + B\gamma(x)\big[\hat{\alpha}(x) + \hat{\gamma}^{-1}(x)v(t) - \alpha(x)\big] \\
&= Az + B\big[v(t) + \Delta v(t)\big],
\end{aligned}
$$

where $\Delta v(t) = \gamma(x)[\hat{\alpha}(x) - \alpha(x) + (\hat{\gamma}^{-1}(x) - \gamma^{-1}(x))v(t)]$. Then, the imperfect modeling is equivalent to having an input disturbance in the resulting linear system, therefore we need to deal with imperfect regulation as well. Overall, we will hope that a well designed optimization algorithm will inherit its robustness to the extremum seeking scheme. That is, if the new sequence $\{\hat{x}_k^s\}$ converges to the minimum, or a small neighborhood of it, given that the error e_k is bounded, then the extremum seeking control schemes in Sect. 4.1 will converge to the minimum or a small neighborhood of it as well. In the following, we will present the robustness analysis of two types of unconstrained optimization algorithm, which provide the foundation of robust design of extremum seeking control.

4.2.1 Robustness of Line Search Methods

Theorem 4.2.1 *Let $J : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable on $\mathbb{R}^n$ and be bounded below. Suppose that ∇J is Lipschitz continuous with constant L. A line search method starting from $\hat{x}_0^s = x_0^s$ is used but with bounded error at each iteration, i.e., $\hat{x}_{k+1}^s = x_{k+1}^s + e_{k+1}$ and $x_{k+1}^s = \hat{x}_k^s + \alpha_k p_k$. Then, the new sequence $\{\hat{x}_k^s\}$ is a descent sequence, that is,*

$$J\big(\hat{x}_{k+1}^s\big) < J\big(\hat{x}_k^s\big),$$

provided

$$\|e_{k+1}\| < \frac{(c\|\nabla J(\hat{x}_k^s)\|^2 \cos^2\theta_k)/L}{\sqrt{\|\nabla J(x_{k+1}^s)\|^2 + c\|\nabla J(\hat{x}_k^s)\|^2\cos^2\theta_k} + \|\nabla J(x_{k+1}^s)\|}, \tag{4.17}$$

where $c = 1$ for exact line search, and $c = 2c_1(1 - c_2)$ for inexact line search satisfying conditions (2.4) *and* (2.5).

Proof Now, for line search method at step $k + 1$, we have

$$\begin{aligned}
J\big(\hat{x}_{k+1}^s\big) - J\big(\hat{x}_k^s\big) &= J\big(x_{k+1}^s + e_{k+1}\big) - J\big(\hat{x}_k^s\big) \\
&\le J\big(x_{k+1}^s\big) + \nabla J\big(x_{k+1}^s\big)^\top e_{k+1} + \frac{L}{2}\|e_{k+1}\|^2 - J\big(\hat{x}_k^s\big), \\
&\le \nabla J\big(x_{k+1}^s\big)^\top e_{k+1} + \frac{L}{2}\|e_{k+1}\|^2 - \frac{c}{2L}\big\|\nabla J\big(\hat{x}_k^s\big)\big\|^2\cos^2\theta_k \\
&\le \frac{L}{2}\bigg[\|e_{k+1}\|^2 + 2\frac{\|\nabla J(x_{k+1}^s)\|}{L}\|e_{k+1}\| - \frac{c}{L^2}\big\|\nabla J\big(\hat{x}_k^s\big)\big\|^2\cos^2\theta_k\bigg] \\
&= \frac{L}{2}\bigg[\bigg(\|e_{k+1}\| + \frac{\|\nabla J(x_{k+1}^s)\|}{L}\bigg)^2 \\
&\quad - \frac{1}{L^2}\bigg(\big\|\nabla J\big(x_{k+1}^s\big)\big\|^2 + c\big\|\nabla J\big(\hat{x}_k^s\big)\big\|^2\cos^2\theta_k\bigg)\bigg],
\end{aligned}$$

where the second line is obtained via the Descent Lemma 2.2.8, and the third line is achieved based on Lemma 2.2.9. Then, if we have

$$\begin{aligned}\|e_{k+1}\| &< \frac{\sqrt{\|\nabla J(x_{k+1}^s)\|^2 + c\|\nabla J(\hat{x}_k^s)\|^2 \cos^2\theta_k} - \|\nabla J(x_{k+1}^s)\|}{L}\\ &= \frac{(c\|\nabla J(\hat{x}_k^s)\|^2 \cos^2\theta_k)/L}{\sqrt{\|\nabla J(x_{k+1}^s)\|^2 + c\|\nabla J(\hat{x}_k^s)\|^2 \cos^2\theta_k} + \|\nabla J(x_{k+1}^s)\|},\end{aligned}$$

we can obtain $J(\hat{x}_{k+1}^s) - J(\hat{x}_k^s) < 0$. □

Although the bound (4.17) is very conservative, it can give us some insights into the robustness of line search methods. First, exact line search allows a larger error bound than inexact line search. Second, we can see that the bound is an increasing function of $\|\nabla J(\hat{x}_k^s)\|$. That is, when $\hat{x}_k^s$ is far away from the minimizer of the performance function, we will expect the gradient to be large and thus the error the algorithm can tolerate is also large. This observation implies that line search methods will be very robust until the gradient converges to some invariant set, which is illustrated in the following corollary.

Corollary 4.2.2 *Let $J : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable on $\mathbb{R}^n$ and be bounded below. Suppose ∇J is Lipschitz continuous with constant L. A steepest descent algorithm is used with bounded error at each iteration. Assuming $\|e_k\| \le e_L$ for some constant e_L, then the gradient of the sequence $\{\hat{x}_k^s\}$ converges to the invariant set satisfying*

$$\|\nabla J(\hat{x}_k^s)\| \le \frac{Le_L}{c}\left[\sqrt{(1+\alpha_k L)^2 + c} + (1+\alpha_k L)\right], \tag{4.18}$$

where $c = 1$ for exact line search, and $c = 2c_1(1-c_2)$ for inexact line search satisfying conditions (2.4) *and* (2.5).

Proof We have $\cos\theta_k = 1$ for steepest descent algorithm, and from inequality (4.17), provided

$$\frac{1}{L}\left(\sqrt{\|\nabla J(x_{k+1}^s)\|^2 + c\|\nabla J(\hat{x}_k^s)\|^2} - \|\nabla J(x_{k+1}^s)\|\right) > e_L,$$

we will always have $J(\hat{x}_{k+1}^s) < J(\hat{x}_k^s)$. So we can find a conservative bound on $\nabla J(\hat{x}_k^s)$ given the error bound e_L.

For steepest descent method, we have $x_{k+1}^s = \hat{x}_k^s - \alpha_k \nabla J(\hat{x}_k^s)$. Then, from the Lipschitz condition it follows that

$$\begin{aligned}\|\nabla J(x_{k+1}^s)\| &\le \|\nabla J(x_{k+1}^s) - \nabla J(\hat{x}_k^s)\| + \|\nabla J(\hat{x}_k^s)\|\\ &\le (1+\alpha_k L)\|\nabla J(\hat{x}_k^s)\|.\end{aligned}$$

Now the bound can be found via

$$\begin{aligned}
&\frac{1}{L}\left(\sqrt{\left\|\nabla J\left(x_{k+1}^s\right)\right\|^2+c\left\|\nabla J\left(\hat{x}_k^s\right)\right\|^2}-\left\|\nabla J\left(x_{k+1}^s\right)\right\|\right)\le e_L\\
\Leftrightarrow\quad &\sqrt{\left\|\nabla J\left(x_{k+1}^s\right)\right\|^2+c\left\|\nabla J\left(\hat{x}_k^s\right)\right\|^2}\le Le_L+\left\|\nabla J\left(x_{k+1}^s\right)\right\|\\
\Leftrightarrow\quad &c\left\|\nabla J\left(\hat{x}_k^s\right)\right\|^2\le 2Le_L(1+\alpha_k L)\left\|\nabla J\left(\hat{x}_k^s\right)\right\|+L^2e_L^2\\
\Leftrightarrow\quad &\left[\sqrt{c}\left\|\nabla J\left(\hat{x}_k^s\right)\right\|-\frac{Le_L}{\sqrt{c}}(1+\alpha_k L)\right]^2\le L^2e_L^2\left[(1+\alpha_k L)^2/c+1\right]\\
\Leftrightarrow\quad &\left\|\nabla J\left(\hat{x}_k^s\right)\right\|-\frac{Le_L}{c}(1+\alpha_k L)\le\frac{Le_L}{c}\sqrt{(1+\alpha_k L)^2+c}.
\end{aligned}$$

Thus, it follows that the gradient of the sequence $\{\hat{x}_k^s\}$ converges to the invariant set satisfying

$$\left\|\nabla J\left(\hat{x}_k^s\right)\right\|\le\frac{Le_L}{c}\left[\sqrt{(1+\alpha_k L)^2+c}+(1+\alpha_k L)\right].\qquad\square$$

From the point of view of inequality (4.18), a diminishing step length α_k is preferred later on to decrease the bound of the invariant set. As $\alpha_k\to 0$, the bound converges to

$$\left(\frac{1}{c}+\sqrt{\frac{1}{c^2}+\frac{1}{c}}\right)Le_L.$$

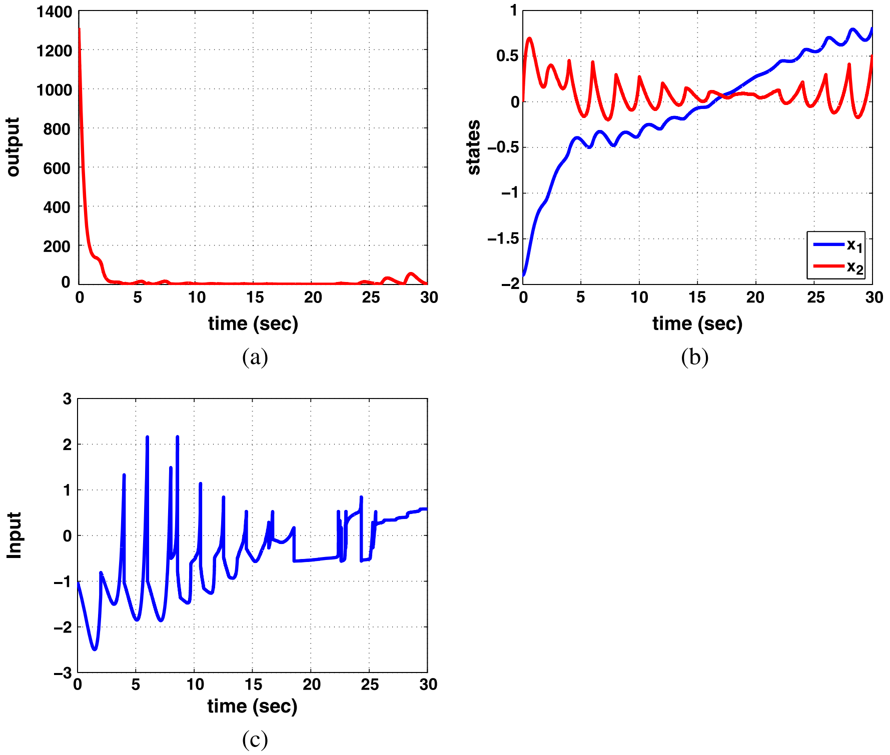
This is consistent with the theory of numerical optimization, where generally a diminishing step length is required for the algorithms to converge to a minimum. And if there is no error between $\hat{x}_k^s$ and x_k^s, we will see that the gradient converges to zero. Moreover, exact line search can achieve a smaller bound than inexact line search.

Example 4.3 Now, we continue the simulation in Sect. 4.1.1 on the LTI system (4.5) with performance function (4.6). A zero-mean random disturbance uniformly distributed with amplitude 0.8 is added to the input. The simulation results are shown in Fig. 4.5 with $x_0^s=[-1.9,0]^\top$, $\delta_k=2$.

The controller (4.3) is unable to precisely regulate the state to the desired set-point. For example, at the first step, the controller cannot transfer the state to the desired destination $x_1^s=[-1.2288,0.1763]^\top$; instead it arrives at $\hat{x}_1^s=[-1.0809,0.2216]^\top$ due to the input disturbance. Then $x_2^s=\hat{x}_1^s-\alpha_k\nabla J(\hat{x}_1^s)=[-0.6771,0.4065]^\top$, that is, the line search method tries to amend the deviated path toward the minimum. Again, the state only arrives at $\hat{x}_2^s=[-0.5289,0.4529]^\top$.

Therefore, eventually we will still have a descent sequence $\{\hat{x}_k^s\}$ as long as the error e_k satisfies the bound (4.17). The comparison of $\{x_k^s\}$ and $\{\hat{x}_k^s\}$ can be seen in Fig. 4.6, where the blue circles represent $\{x_k^s\}$, magenta squares denote $\{\hat{x}_k^s\}$ and the red dashed line is the state trajectory.

Interestingly, we find that the disturbance actually helps the algorithm to achieve more reduction in function values in the first few steps by comparing with the ideal



(a)

(b)

(c)

Fig. 4.5 Extremum seeking control for a LTI system with input disturbance: (**a**) performance output; (**b**) states

case (see Fig. 4.2 and Fig. 4.6). However, as shown in Fig. 4.5(a), the performance output becomes diverging eventually since the state regulator design (4.3) with input disturbance cannot regulate the state even into the desired neighborhood (4.17) of x_{k+1}^s as $\nabla J(x)$ is very small. Therefore, in such cases, we will prefer using (4.18) as the stopping criterion, which is the best we can do given the current design of state regulator (4.3) with input disturbance.

4.2.2 Robustness of Trust Region Methods

For trust region methods, the ratio ρ_k in (2.9), repeated here for convenience,

$$\rho_k = \frac{J(x_k^s) - J(x_k^s + p_k)}{m_k(x_k^s) - m_k(x_k^s + p_k)},$$

is required at each step to determine x_{k+1}^s. Let us call the kth step a successful step if $\rho_k > \eta$, and an unsuccessful step if $\rho_k \leq \eta$. However, in order to obtain ρ_k, we

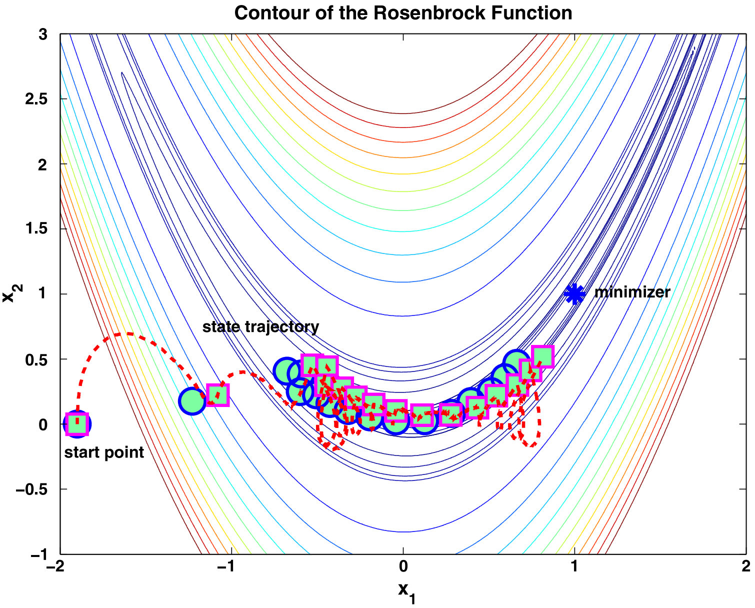


Fig. 4.6 Extremum seeking control for a LTI system with input disturbance: phase portrait of the steepest descent sequence $\{x_k^s\}$ over the contour of the performance function

need the measurement of $J(x_k^s + p_k)$. Then, in the implementation of extremum seeking control based on trust region method, this means we need to regulate the state x to $x_k^s + p_k$ in order to obtain measurement of $J(x_k^s + p_k)$, even if it is not the next iterate x_{k+1}^s. For example, at the kth step, the controller needs to regulate the state $x(t_k)$ to $x_k^s + p_k$ to obtain the ratio ρ_k. If $\rho_k > \eta$, then it is a successful step and we are done. Otherwise, if $\rho \le \eta$, then we need to regulate the state back to x_k^s since it is an unsuccessful step. However, for practical applications, the ideal ratio ρ_k cannot be obtained due to the imperfect regulation. The following theorem provides one possible quantitative analysis for the convergence of trust region methods under bounded error.

Theorem 4.2.3 *Let $J : \mathbb{R}^n \to \mathbb{R}$ be Lipschitz continuously differentiable and bounded below on the level set $\{x|J(x) \le J(x_0^s)\}$. Suppose that ∇J is Lipschitz continuous with constant L. A trust region method starting from $\hat{x}_0^s = x_0^s$ is used, but with a bounded error at each iteration. That is, $\hat{x}_{k+1}^s = \hat{x}_k^s + p_k + e_{k+1}$ for a successful step, or $\hat{x}_{k+1}^s = \hat{x}_k^s + e_{k+1}$ for an unsuccessful step. If, for every successful step*

$$\|e_{k+1}\| \le -\frac{2}{L} \left\| \nabla J\left(\hat{x}_k^s + p_k\right) \right\| \cos \alpha_k, \tag{4.19}$$

and for every unsuccessful step,

$$\|e_{k+1}\| \le -\frac{2}{L} \left\| \nabla J\left(\hat{x}_k^s\right) \right\| \cos \beta_k, \tag{4.20}$$

where

$$\cos\alpha_k = \frac{\nabla J(\hat{x}_k^s + p_k)^\top e_{k+1}}{\|\nabla J(\hat{x}_k^s + p_k)\| \|e_{k+1}\|}$$

and

$$\cos\beta_k = \frac{\nabla J(\hat{x}_k^s)^\top e_{k+1}}{\|\nabla J(\hat{x}_k^s)\| \|e_{k+1}\|},$$

then there exists a subsequence of $\{\hat{x}_k^s\}$ *that converges to the first-order stationary point of* $J(x)$.

Proof At iteration k, $t = t_k$, and we obtain p_k by approximately solving the trust region subproblem (2.8). Ideally, we would use a regulator such as (4.3) to drive the state from $x(t_k) = \hat{x}_k^s$ to $\hat{x}_k^s + p_k$ to obtain an ideal ratio (2.9). However, due to the input disturbance or model uncertainty, such a controller can only regulate the state to $\hat{x}_k^s + p_k + e_{k+1}$ and conclude with a practical ratio

$$\hat{\rho}_k = \frac{J(\hat{x}_k^s) - J(\hat{x}_k^s + p_k + e_{k+1})}{m_k(\hat{x}_k^s) - m_k(\hat{x}_k^s + p_k)}. \tag{4.21}$$

Let $S_k = \{s_1, s_2, \ldots, s_i\}$ be a subsequence of $\{1, 2, \ldots, k\}$ such that S_k represents the index set of successful steps $\{\hat{x}_{s_i}^s\}$ up to $t = t_k$, that is $s_i \le k$ and

$$\frac{J(\hat{x}_{s_{i-1}}^s) - J(\hat{x}_k^s)}{m_{s_{i-1}}(\hat{x}_{s_{i-1}}^s) - m_{s_{i-1}}(\hat{x}_{s_{i-1}}^s + p_{s_{i-1}})} \ge \eta, \tag{4.22}$$

which means $\hat{x}_k^s$ is a successful step if started from $\hat{x}_{s_{i-1}}^s$. Two cases need to be analyzed to guarantee the global convergence of the sequence $\{\hat{x}_k^s\}$.

Case I: For a successful step, we have $\rho_k > \eta$. Therefore, we want $\hat{\rho}_k > \eta$ to guarantee a successful step in the presence of e_{k+1}. That is, we wish

$$\begin{aligned}
&\frac{J(\hat{x}_k^s) - J(\hat{x}_k^s + p_k + e_{k+1})}{m_k(\hat{x}_k^s) - m_k(\hat{x}_k^s + p_k)} \\
&\ge \frac{J(\hat{x}_k^s) - J(\hat{x}_k^s + p_k) - \nabla J(\hat{x}_k^s + p_k)^\top e_{k+1} - \frac{L}{2}\|e_{k+1}\|^2}{m_k(\hat{x}_k^s) - m_k(\hat{x}_k^s + p_k)} \\
&= \frac{J(\hat{x}_k^s) - J(\hat{x}_k^s + p_k)}{m_k(\hat{x}_k^s) - m_k(\hat{x}_k^s + p_k)} - \frac{\nabla J(\hat{x}_k^s + p_k)^\top e_{k+1} + \frac{L}{2}\|e_{k+1}\|^2}{m_k(\hat{x}_k^s) - m_k(\hat{x}_k^s + p_k)} \\
&> \eta - \frac{\nabla J(\hat{x}_k^s + p_k)^\top e_{k+1} + \frac{L}{2}\|e_{k+1}\|^2}{m_k(\hat{x}_k^s) - m_k(\hat{x}_k^s + p_k)} \\
&\ge \eta.
\end{aligned}$$

Since $m_k(\hat{x}_k^s) - m_k(\hat{x}_k^s + p_k)$ is always positive, we need

$$\nabla J\big(\hat{x}_k^s + p_k\big)^\top e_{k+1} + \frac{L}{2}\|e_{k+1}\|^2 \le 0.$$

Now define α_k to be the angle between $\nabla J(\hat{x}_k^s + p_k)$ and e_{k+1}, implying that

$$\cos\alpha_k = \frac{\nabla J(\hat{x}_k^s + p_k)^\top e_{k+1}}{\|\nabla J(\hat{x}_k^s + p_k)\|\,\|e_{k+1}\|}.$$

Then,

$$\begin{aligned}
&\nabla J\big(\hat{x}_k^s + p_k\big)^\top e_{k+1} + \frac{L}{2}\|e_{k+1}\|^2 \\
&\quad = \big\|\nabla J\big(\hat{x}_k^s + p_k\big)\big\|\,\|e_{k+1}\|\cos\alpha_k + \frac{L}{2}\|e_{k+1}\|^2 \le 0 \\
&\quad \Leftrightarrow \quad \|e_{k+1}\| \le -\frac{2}{L}\big\|\nabla J\big(\hat{x}_k^s + p_k\big)\big\|\cos\alpha_k.
\end{aligned}$$

Case II: For an unsuccessful step, we have $\rho_k \le \eta$. This means that we do not need to care about the regulation error to the set-point $\hat{x}_k^s + p_k$. If the resulting $\hat{\rho}_k > \eta$, it means that the state has been regulated to a point with lower performance function value, which actually helps the optimization process. If $\hat{\rho}_k \le \eta$, then we need to regulate the state back to $\hat{x}_k^s$. Here, we redefine the regulation error e_{k+1} as an error from this second state regulation, i.e., $\hat{x}_{k+1}^s = \hat{x}_k^s + e_{k+1}$. Therefore, we wish $\hat{x}_k^s + e_{k+1}$ to be a successful step compared with the previous one $\hat{x}_{s_{i-1}}^s$, i.e.,

$$\begin{aligned}
&\frac{J(\hat{x}_{s_{i-1}}^s) - J(\hat{x}_k^s + e_{k+1})}{m_{s_{i-1}}(\hat{x}_{s_{i-1}}^s) - m_{s_{i-1}}(\hat{x}_{s_{i-1}}^s + p_{s_{i-1}})} \\
&\quad \ge \frac{J(\hat{x}_{s_{i-1}}^s) - J(\hat{x}_k^s) - \nabla J(\hat{x}_k^s)^\top e_{k+1} - \frac{L}{2}\|e_{k+1}\|^2}{m_{s_{i-1}}(\hat{x}_{s_{i-1}}^s) - m_{s_{i-1}}(\hat{x}_{s_{i-1}}^s + p_{s_{i-1}})} \\
&\quad = \frac{J(\hat{x}_{s_{i-1}}^s) - J(\hat{x}_k^s)}{m_{s_{i-1}}(\hat{x}_{s_{i-1}}^s) - m_{s_{i-1}}(\hat{x}_{s_{i-1}}^s + p_{s_{i-1}})} - \frac{\nabla J(\hat{x}_k^s)^\top e_{k+1} + \frac{L}{2}\|e_{k+1}\|^2}{m_{s_{i-1}}(\hat{x}_{s_{i-1}}^s) - m_{s_{i-1}}(\hat{x}_{s_{i-1}}^s + p_{s_{i-1}})} \\
&\quad > \eta - \frac{\nabla J(\hat{x}_k^s)^\top e_{k+1} + \frac{L}{2}\|e_{k+1}\|^2}{m_{s_{i-1}}(\hat{x}_{s_{i-1}}^s) - m_{s_{i-1}}(\hat{x}_{s_{i-1}}^s + p_{s_{i-1}})} \\
&\quad \ge \eta.
\end{aligned}$$

Now define β_k to be the angle between $\nabla J(\hat{x}_k^s)$ and e_{k+1}, that is,

$$\cos\beta_k = \frac{\nabla J(\hat{x}_k^s)^\top e_{k+1}}{\|\nabla J(\hat{x}_k^s)\|\,\|e_{k+1}\|}.$$

Then it follows that

$$\nabla J\left(\hat{x}_k^s\right)^\top e_{k+1} + \frac{L}{2}\|e_{k+1}\|^2 = \left\|\nabla J\left(\hat{x}_k^s\right)\right\| \|e_{k+1}\| \cos\beta_k + \frac{L}{2}\|e_{k+1}\|^2 \leq 0$$

$$\Leftrightarrow \quad \|e_{k+1}\| \leq -\frac{2}{L}\left\|\nabla J\left(\hat{x}_k^s\right)\right\| \cos\beta_k.$$

Thus, if (4.19) and (4.20) are satisfied, we conclude that we will have a descent subsequence $\{\hat{x}_{s_k}\}$ of $\{\hat{x}_k^s\}$ globally converging to the first-order stationary point of $J(x)$. □

4.2.3 Robust Extremum Seeking Control Design

The importance of Theorems 4.2.1 and 4.2.3 is that they relax the design requirements for the state regulator, thus removing the need for perfect state regulation to x_{k+1}^s. Robust design of a state regulator to deal with disturbances, unmodeled plant dynamics via adaptive control [8] or sliding mode control can be made easier by regulating state x to a neighborhood of x_{k+1}^s, which will be presented in the next chapter.

For extremum seeking control based on line search methods, as long as we can design a regulator satisfying (4.17), then the extremum seeking will continue to decrease the performance output.

For extremum seeking control based on trust region methods, at time $t = t_k$, we do not know whether it will be a successful step or not in advance, since we do not have the knowledge of ρ_k. In this case, one way to guarantee the convergence under input disturbance is the following. The state regulator for set-point $x_k^s + p_k$ needs to be designed to satisfy (4.19). In the case of a successful step, then the design will guarantee the step still to be a successful step since $\hat{\rho}_k > \eta$ (see (4.21) in Sect. 4.2.2). If it is not a successful step, that is, $\rho_k < \eta$, then there is no guarantee that $\hat{\rho}_k$ will be greater than η. If $\hat{\rho}_k > \eta$, then it means that the controller is on the right way due to the disturbance. If not, then we need to regulate the state back to $\hat{x}_k^s$; that is, we need to design the regulator for $\hat{x}_k^s$ to satisfy (4.20), which means we can return to the right path as inferred from previous steps.

Example 4.4 Both inequalities (4.19) and (4.20) define similar criteria for the design of state regulators. A pictorial illustration of such criteria in $\mathbb{R}^2$ is shown in Fig. 4.7, where we let the set-point, $\hat{x}_k^s + p_k$ in (4.19) and $\hat{x}_k^s$ in (4.20), be $[0, 0]^\top$. We also let its gradient be $[10, 10]^\top$, and let the Lipschitz constant be $L = 2$. The blue star is the desired destination, and the dotted area depicts the acceptable region satisfying inequalities (4.19) and (4.20). The size of the region is proportional to the gradient of the set-point, and the reciprocal of the Lipschitz constant.

4.3 Conclusion

In this chapter, we successfully incorporate numerical optimization algorithms into the set up of an extremum seeking control scheme. The convergence of the pro-

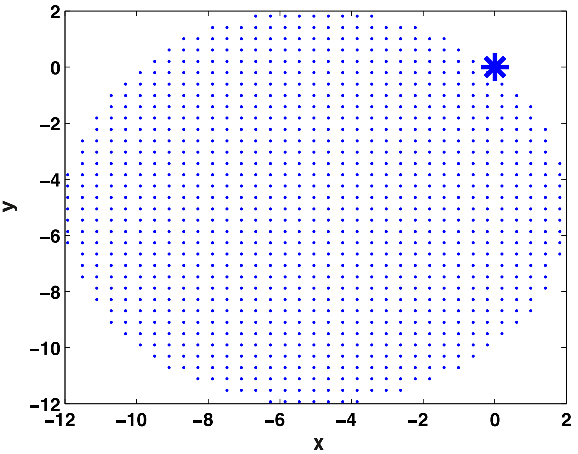


Fig. 4.7 Design criterion for the robust regulator to satisfy (4.19) and (4.20) of the trust region based extremum seeking control

posed extremum seeking control scheme is guaranteed if the optimization algorithm is globally convergent and with appropriate state regulation. We also analyze the robustness of line search methods and trust region methods, which relaxes the design requirement for the state regulator and provides further flexibility in designing the robust extremum seeking control scheme. Numerical examples are given to illustrate the analysis results.

The current setting of the proposed extremum seeking control scheme is one attractive way to use the numerical optimization for the purpose of real time optimization, as it retains the global convergence property of the numerical optimization algorithm. It allows us to utilize the research results from the optimization community. For example, when $D \subset \mathbb{R}^n$, it becomes a constrained extremum seeking control problem, and generally we need to resort first to constrained optimization algorithms. In this case, the state regulator design will be more challenging as it needs to ensure that the state will not violate the constraints during the transient. The exploration of more robust numerical optimization algorithms and the design of robust state regulators will be two ways to enhance the robustness of the extremum seeking control schemes. Moreover, the design of the regulation time δ_k needs to be further studied to deal with various requirements originating from practical applications.

References

1. Chen, C.-T.: Linear System Theory and Design. Oxford University Press, New York (1999)
2. Conn, A.R., Scheinberg, K., Toint, P.L.: Recent progress in unconstrained nonlinear optimization without derivatives. Math. Program. **79**, 397–414 (1997)
3. Freund, R.: Lecture Notes of 15.084j/6.252j Nonlinear Programming (Spring 2004). MIT OpenCourseWare
4. Khalil, H.K.: Nonlinear Systems. Prentice Hall, Upper Saddle River (2001)
5. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev. **45**(3), 385–482 (2003)
6. Nocedal, J., Wright, S.: Numerical Optimization. Springer, Berlin (1999)

7. Powell, M.J.D.: UOBYQA: Unconstrained optimization by quadratic approximation. Technical Report NA2000/14, DAMTP, University of Cambridge (2000)
8. Spooner, J.T., Maggiore, M., Ordóñez, R., Passino, K.M.: Stable Adaptive Control and Estimation for Nonlinear Systems, Neural and Fuzzy Approximator Techniques. Wiley, New York (2002)
9. Zhang, C., Ordóñez, R.: Non-gradient extremum seeking control of feedback linearizable systems with application to ABS design. In: Proceedings of the Conference Decision and Control, pp. 6666–6671 (2006)
10. Zhang, C., Ordóñez, R.: Numerical optimization-based extremum seeking control with application to ABS design. IEEE Trans. Autom. Control **52**(3), 454–467 (2007)

Chapter 5
Asymptotic State Regulator Design

5.1 Problem Statement

In this chapter, we focus on minimum seeking control of the SISO nonlinear affine system given by

$$\dot{x} = f(x) + g(x)u, \tag{5.1}$$

and the performance output

$$y = J(x), \tag{5.2}$$

where we assume[1] $D = \mathbb{R}^n$. Notice that this is a rather broad class of systems, containing a large number of systems of engineering and scientific interest. In addition to Assumptions 3.2.1, 3.2.2 and 3.2.3, we assume

Assumption 5.1.1 *The performance function $J(x)$ is convex on the domain D.*

Assumption 5.1.1 means the stationary point condition becomes a necessary and sufficient condition to identify a global minimizer. Optimization algorithms with global convergence are then able to converge to a global minimum x^*. Without such assumption, we just reduce the convergence to a stationary point, and the validity of all the results to follow is preserved. The goal of ESC is to design a controller based on output measurements and state measurements to regulate the state to an unknown minimizer x^* of an unknown performance function $J(x)$.

Assume the nonlinear affine system (5.1) is state feedback linearizable on the domain D, as in Assumption 4.1.12. Then, there exists a diffeomorphism $T : D \to \mathbb{R}^n$ such that $D_z = T(D)$ contains the origin and the change of variables $z = T(x)$ transforms the system (5.1) into the form (see [2] for more details)

$$\dot{z} = Az + B\big(f(z) + g(z)u\big) \tag{5.3}$$

[1] Restrictions on D will result in the need for constrained optimization. The stability of constrained ESC is still an open topic of research.

C. Zhang, R. Ordóñez, *Extremum-Seeking Control and Applications*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2224-1_5, © Springer-Verlag London Limited 2012

with (A, B) in controllable canonical form (4.10), and $g(z) = g(T(x))$ non-singular for all $x \in D$. Note that we assume $D = \mathbb{R}^n$; therefore, we need to design a state regulator u for the state feedback linearizable system (5.3) that drives the state to x^s_{k+1}. This is so as to obtain the necessary measurement of the performance output, or its gradient, at $x = x^s_{k+1}$, needed in order to continue the optimization process, as stated in Step 3 of the algorithm shown in Sect. 3.2.

We have proposed a design method for a finite-time state regulator for system (5.3) in Sect. 4.1.2 of Chap. 4. This design approach is based on the controllability proof of LTI systems. As shown in Chap. 4, the finite-time regulator can transfer the state from $x(t_k)$ to x^s_{k+1} in a predefined δ_k time. The convergence result is stated in Theorem 4.1.13. However, input disturbances and unmodeled plant dynamics can result in imperfect regulation of the finite-time state regulator, by driving the state only to a neighborhood of the set point x^s_{k+1}. This regulation error can result in the closed-loop system becoming unstable. Theorems 4.2.1 and 4.2.3 provide criteria stating that if the regulation neighborhood is always within the robust region of the optimization algorithm used in Step 1 of the NOESC scheme, then extremum seeking can still be achieved. However, these theorems do not illustrate a methodology to design a robust NOESC loop. In fact, it is generally very challenging to robustify the finite-time state regulator, since it is inherently an open-loop approach between initial and final conditions.

Moreover, the finite-time state regulator in (4.11) can only guarantee that the state will arrive at the desired x^s_{k+1} at the prescribed time (of course, as long as that the system is controllable); however, this regulator does not guarantee the system will stay at the set point after that.

Example 5.1 As a simple illustration of this issue, consider a point mass model given by

$$m\ddot{z} = u,$$

where $z \in \mathbb{R}$ is the position of the mass, and we set $m = 1$ for simplicity. Letting $x_1 = z$, $x_2 = \dot{z}$, the state variable model is

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u.$$

It is easy to verify that the point mass model is controllable, and therefore we can design a finite-time state regulator to drive the state to an arbitrary point in the state space. For example, we start the point mass at the initial condition $x^s_0 = [0, 0]^\top$, and we want to drive the point mass to the state $[1, 2]^\top$. The finite-time state regulator can guarantee the state trajectory will pass through the point $[1, 2]^\top$. However, it is clear that the point mass cannot remain in this state, since the velocity state is 2. That is, $[1, 2]^\top$ is not an equilibrium point. The point mass state can only stay at an equilibrium point, such as $[c, 0]^\top$ for any $c \in \mathbb{R}$. Of course, for some systems, any point in the state space is a stabilizable equilibrium. But then, requiring this condition would be similar to the assumptions in Chap. 5 of [1], where the performance function is assumed to be reducible to a function of a single parameter of the control law, and then it is assumed that this parameter is sufficient to produce a stabilizable equilibrium point.

As this example shows, even if the optimization algorithm can find the extremum x^* for regulation, the controller (4.11) cannot guarantee that x^* can be converted into an equilibrium point, and consequently, that the system will operate at x^* permanently. It is for this reason that we observe an oscillating steady state performance in Fig. 4.3 of Example 4.2, where the optimizer keeps commanding $x^* = [1, 1]^\top$ as the destination, but the best the controller can do is to enable the state to visit x^* every δ_K time to create an oscillating behavior. Thus we require Assumption 3.2.3 to preclude an oscillating steady state behavior.

In order to address these shortcomings of the basic NOESC method introduced in Chap. 4, in the following sections we will present a new design of state regulator via asymptotic tracking, which provides the basis for robust NOESC design.

5.2 Asymptotic State Regulator Design for NOESC

Inspired by the standard output-tracking problem [6], we will design an asymptotic state regulator that regulates the state x asymptotically to x^s_{k+1}. Let $z = T(x)$, where T is a diffeomorphism in D. We will use the following notation: x_i, z_i are the ith coordinates of the state vectors x and z, respectively. Moreover, x^s_{k+1} and $z^s_{k+1} = T(x^s_{k+1})$ are the set points generated by the optimization algorithm in x and z coordinates, respectively, at the $(k+1)$th step; then, $x^s_{k+1,i}$, $z^s_{k+1,i}$ denote the ith coordinates of the set points x^s_{k+1} and z^s_{k+1}, respectively.

Given a bounded reference signal $r_k(t) \in \mathbb{R}$, let

$$e_z = k_1(z_1 - r_k) + \cdots + k_{n-1}\bigl(z_{n-1} - r_k^{(n-2)}\bigr) + z_n - r_k^{(n-1)}, \tag{5.4}$$

where the polynomial

$$s^{n-1} + k_{n-1}s^{n-2} + \cdots + k_1$$

is Hurwitz, and we define the auxiliary signal

$$\chi(z) = k_1(z_2 - \dot{r}_k) + \cdots + k_{n-1}\bigl(z_n - r_k^{(n-1)}\bigr) - r_k^{(n)}.$$

Thus, the control design

$$u = \frac{-\chi(z) - f(z) - k_u e_z}{g(z)}, \tag{5.5}$$

with $k_u > 0$, will ensure $e_z \to 0$ and hence $z_1 \to r_k(t)$, $z_2 \to \dot{r}_k(t), \ldots$ for the state feedback linearizable system (5.3). From [6], we have

$$\|z - \bar{r}_k\| \le k_{\sum}\phi_\mu\bigl(t, |e_z|\bigr) + |e_z|, \tag{5.6}$$

where

$$k_{\sum} = \big(1 + |k_1| + \cdots + |k_{n-1}|\big), \tag{5.7}$$

$$\bar{r}_k = \big[r_k, \dot{r}_k, \ldots, r_k^{(n-1)}\big]^\top, \tag{5.8}$$

$$\mu = \big[z_1 - r_k, \ldots, z_{n-1} - r_k^{(n-2)}\big]^\top, \tag{5.9}$$

and

$$\begin{aligned}\big\|\mu(t)\big\| &\le \phi_\mu\big(t, |e_z|\big)\\ &= d_1 e^{-d_2 t}\big\|\mu(0)\big\| + \frac{d_1|e_z(0)|}{d_2 - k_u}\big(e^{-k_u t} - e^{-d_2 t}\big)\end{aligned} \tag{5.10}$$

for some positive scalars d_1 and $d_2 \neq k_u$. We have $\phi_\mu(t, |e_z|)$ bounded for any bounded e_z, nondecreasing with respect to $|e_z| \in \mathbb{R}^+$ for each fixed t, and decreasing to zero as $t \to \infty$. Now, we present the Theorem for the asymptotic state regulator design.

Theorem 5.2.1 *Assume the nonlinear system* (5.1) *is state feedback linearizable on the domain D, with T the diffeomorphism that transforms the system to normal form* (5.3), *and $\|\frac{\partial T^{-1}}{\partial x}\|$ is bounded on D. At time $t = t_k$, with state $x = x(t_k)$, the desired set point is x^s_{k+1}. The reference trajectory $r_k(t)$ is designed to be a bounded periodic signal satisfying the condition*

$$\begin{aligned} r_k(mT_r) &= z^s_{k+1,1},\\ \dot{r}_k(mT_r) &= z^s_{k+1,2},\\ &\vdots\\ r_k^{(n-1)}(mT_r) &= z^s_{k+1,n},\end{aligned} \tag{5.11}$$

where $z^s_{k+1} = T(x^s_{k+1}), m$ is a positive integer, and $T_r > 0$ is the period of $r_k(t)$. By applying the controller (5.5), *we will have x asymptotically tracking $T^{-1}(\bar{r}_k)$, which will visit the desired set point x^s_{k+1} every T_r time.*

Proof As seen above, the controller (5.5) implies z_1 will asymptotically track $r_k(t)$. Since T is a diffeomorphism on D, then T^{-1} is continuously differentiable. Because we also assume $\|\frac{\partial T^{-1}}{\partial x}\|$ is bounded on D, then T^{-1} is Lipschitz continuous on D for some Lipschitz constant L_T. Now, we have

$$\begin{aligned}\big\|x - T^{-1}(\bar{r}_k)\big\| &= \big\|T^{-1}(z) - T^{-1}(\bar{r}_k)\big\|\\ &\le L_T\|z - \bar{r}_k\|\\ &\le L_T\big(|e_z| + k_{\sum}\phi_\mu\big(t, |e_z|\big)\big).\end{aligned} \tag{5.12}$$

By virtue of controller (5.5), and from (5.6) and (5.10), e_z and $\phi_\mu(t, |e_z|)$ will globally converge to zero as $t \to \infty$, therefore we will have x asymptotically tracking $T^{-1}(\bar{r}_k)$. Moreover, from condition (5.11), we have $\bar{r}_k(mT_r) = z^s_{k+1}$ and therefore

$$\begin{aligned} \left\| T^{-1}\big(\bar{r}_k(t)\big) - x^s_{k+1} \right\| &= \left\| T^{-1}\big(\bar{r}_k(t)\big) - T^{-1}\big(z^s_{k+1}\big) \right\| \\ &\le L_T \left\| \bar{r}_k(t) - z^s_{k+1} \right\| \\ &= 0, \end{aligned} \tag{5.13}$$

at $t = mT_r$, where m is an arbitrary positive integer. This completes the proof. □

Theorem 5.2.1 converts the state regulation problem $x \to x^s_{k+1}$ into the error regulation problem $e_z \to 0$, where the system dynamics can be written as

$$\dot{e}_z = \chi(z) + f(z) + g(z)u.$$

This enables the robust ESC design. One possible design of reference signal $r_k(t)$ satisfying (5.11) can be found below.

5.2.1 Construction of Reference Signal

Here, we will illustrate one possible approach to constructing the bounded periodic reference signal satisfying condition (5.11). First, assume that the number of states n is odd and let $p = (n-1)/2$. Then, choose

$$\begin{aligned} r_k(t) = {} & a_1 \sin(\omega_1 t) + b_1 \cos(\omega_1 t) + a_2 \sin(\omega_2 t) + b_2 \cos(\omega_2 t) + \cdots \\ & + a_p \sin(\omega_p t) + b_p \cos(\omega_p t) + r_{k0}, \end{aligned} \tag{5.14}$$

where a_i, b_i, ω_i and r_{k0} for $i = 1, \dots, p$ are parameters to be determined based on (5.11). We have

$$\begin{aligned} \dot{r}_k(t) &= a_1\omega_1 \cos(\omega_1 t) - b_1\omega_1 \sin(\omega_1 t) + \cdots \\ &\quad + a_p\omega_p \cos(\omega_p t) - b_p\omega_p \sin(\omega_p t), \\ \ddot{r}_k(t) &= -a_1\omega_1^2 \sin(\omega_1 t) - b_1\omega_1^2 \cos(\omega_1 t) + \cdots \\ &\quad - a_p\omega_p^2 \sin(\omega_p t) - b_p\omega_p^2 \cos(\omega_p t), \\ &\ \ \vdots \\ r_k^{(2p-1)}(t) &= (-1)^{p-1} a_1\omega_1^{2p-1} \cos(\omega_1 t) \\ &\quad - (-1)^{p-1} b_1\omega_1^{2p-1} \sin(\omega_1 t) + \cdots \\ &\quad + (-1)^{p-1} a_p\omega_p^{2p-1} \cos(\omega_p t) \end{aligned} \tag{5.15}$$

$$\begin{aligned}
&- (-1)^{p-1} b_p \omega_p^{2p-1} \sin(\omega_p t),\\
r_k^{(2p)}(t) &= (-1)^p a_1 \omega_1^{2p} \sin(\omega_1 t)\\
&+ (-1)^p b_1 \omega_1^{2p} \cos(\omega_1 t) + \cdots\\
&+ (-1)^p a_p \omega_p^{2p} \sin(\omega_p t)\\
&+ (-1)^p b_p \omega_p^{2p} \cos(\omega_p t).
\end{aligned}$$

Then, since T_r is the period of the signal $r_k(t)$, such that $\omega_i T_r = 2m_i\pi$ for some positive integers m_i, $i = 1, \ldots, p$, we have

$$\begin{aligned}
r_k(mT_r) &= r_k(T_r)\\
&= b_1 + b_2 + \cdots + b_p + r_{k0}\\
&= z_{k+1,1}^s\\
\dot{r}_k(mT_r) &= \dot{r}_k(T_r)\\
&= a_1\omega_1 + a_2\omega_2 + \cdots + a_p\omega_p\\
&= z_{k+1,2}^s\\
\ddot{r}_k(mT_r) &= \ddot{r}_k(T_r)\\
&= -b_1\omega_1^2 - b_2\omega_2^2 - \cdots - b_p\omega_p^2\\
&= z_{k+1,3}^s\\
&\vdots\\
r_k^{(2p-1)}(mT_r) &= r_k^{(2p-1)}(T_r)\\
&= (-1)^{p-1}\big[a_1\omega_1^{2p-1} + \cdots + a_p\omega_p^{2p-1}\big]\\
&= z_{k+1,2p}^s\\
r_k^{(2p)}(mT_r) &= r_k^{(2p)}(T_r)\\
&= (-1)^p\big[b_1\omega_1^{2p} + \cdots + b_p\omega_p^{2p}\big]\\
&= z_{k+1,2p+1}^s.
\end{aligned}$$

Thus, we can solve the linear equations

$$\underbrace{\begin{bmatrix} \omega_1 & \omega_2 & \cdots & \omega_p \\ \omega_1^3 & \omega_2^3 & \cdots & \omega_p^3 \\ \vdots & \vdots & \ddots & \vdots \\ \omega_1^{2p-1} & \omega_2^{2p-1} & \cdots & \omega_p^{2p-1} \end{bmatrix}}_{\Omega_a} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} (-1)^{1-1} z_{k+1,2}^s \\ (-1)^{2-1} z_{k+1,4}^s \\ \vdots \\ (-1)^{p-1} z_{k+1,2p}^s \end{bmatrix}, \tag{5.16}$$

$$\underbrace{\begin{bmatrix} \omega_1^2 & \omega_2^2 & \cdots & \omega_p^2 \\ \omega_1^4 & \omega_2^4 & \cdots & \omega_p^4 \\ \vdots & \vdots & \ddots & \vdots \\ \omega_1^{2p} & \omega_2^{2p} & \cdots & \omega_p^{2p} \end{bmatrix}}_{\Omega_b} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{bmatrix} = \begin{bmatrix} (-1)^1 z_{k+1,3} \\ (-1)^2 z_{k+1,5} \\ \vdots \\ (-1)^p z_{k+1,2p+1} \end{bmatrix} \tag{5.17}$$

in order to obtain the parameters a_i, b_i, ω_i and r_{k0}. We can first choose different frequencies as $0 < \omega_1 < \omega_2 < \cdots < \omega_n$, then the generalized Vandermonde matrices Ω_a in (5.16) and Ω_b in (5.17) will have positive determinants and hence are nonsingular, so that we can obtain unique a_i, b_i for $i = 1, \dots, p$ by solving (5.16) and (5.17). Finally, we can solve

$$r_{k0} = z_{k+1,1}^s - \sum_{i=1}^{p} b_p.$$

Finally, if the number of states is even and greater than 2, we can let $p = n/2$ and $r_k(t)$ be (5.14) with $r_{k0} = 0$, then we will have a very similar result as in the case when n is odd.

5.2.2 *Using the Asymptotic State Regulator in Finite Time*

The asymptotic state regulator in Theorem 5.2.1 may take infinite time in order for the state x to reach the required set point x_{k+1}^s. For this reason, convergence results of NOESC using a finite-time state regulator as in Theorem 4.1.13 do not directly apply here. However, certain optimization algorithms are still functional as long as we can regulate the state to a neighborhood of the set point x_{k+1}^s. Therefore, we may still be able to implement the asymptotic state regulator (5.5) for a finite time and guarantee the convergence given a robust optimization algorithm is used and some conditions are met.

In particular, recall Theorem 4.2.1, which studies convergence of line search methods under bounded errors. This theorem can be used for our purposes here: It implies that given the current state $x = \hat{x}_k^s$, if we use the line search method to generate the new set point $x_{k+1}^s = \hat{x}_k^s + \alpha_k p_k$, we can still ensure the sequence $\{\hat{x}_k^s\}$ is a descent sequence and the NOESC is convergent, even in the presence of regulation errors. This is true as long as the control (5.5) drives the state into the neighborhood of x_{k+1}^s given by (4.17). Indeed, the asymptotic state regulator (5.5) can drive the state into the desired neighborhood (4.17) of x_{k+1}^s in a finite time δ_k, which can be estimated in advance as shown in the following result.

Theorem 5.2.2 *Assume the nonlinear system* (5.1) *is state feedback linearizable on the domain D, with T the diffeomorphism that transforms the system to normal form* (5.3), *and $\| \frac{\partial T^{-1}}{\partial x} \|$ is bounded on D. At time $t = t_k$ we have $x = x(t_k) = \hat{x}_k^s$, and the desired set point $x_{k+1}^s = \hat{x}_k^s + \alpha_k p_k$ is generated by a line search algorithm. The* *asymptotic state regulator* (5.5) *will drive the state into the neighborhood* (4.17) *of* x_{k+1}^s *in* δ_k *time. An upper bound of the regulation time is*

$$\bar{\delta}_k = m_k T_r, \tag{5.18}$$

where T_r *is the period of the reference trajectory* $r_k(t)$, *and* m_k *is the smallest integer satisfying*

$$m_k T_r > \frac{1}{\min(k_u, d_2)} \ln\left(\frac{k_s L_T}{b_k \|\nabla J(\hat{x}_k^s)\|}\right), \tag{5.19}$$

where

$$k_s = \left|e_z(0)\right| + k_{\sum}\left(d_1 \left\|\mu(0)\right\| + \frac{d_1|e_z(0)|}{|d_2 - k_u|}\right),$$

$$b_k = \frac{cc_3^2}{L(\sqrt{(1+\alpha_k L)^2 + c} + (1+\alpha_k L))}.$$

The constant $k_{\sum}$ *is defined in* (5.7), d_1 *and* d_2 *are defined in* (5.10), *and* k_u *is defined in* (5.5).

Proof Since T is a diffeomorphism on D, then T^{-1} is continuously differentiable. Because we also assume $\|\frac{\partial T^{-1}}{\partial x}\|$ is bounded on D, then T^{-1} is Lipschitz continuous on D for some Lipschitz constant L_T. Since we have no measurement of $\nabla J(x_{k+1}^s)$ unless we drive the state to x_{k+1}^s, the right hand side of inequality (4.17), shown here again for convenience, is not known yet:

$$\|e_{k+1}\| < \frac{(c\|\nabla J(\hat{x}_k^s)\|^2 \cos^2\theta_k)/L}{\sqrt{\|\nabla J(x_{k+1}^s)\|^2 + c\|\nabla J(\hat{x}_k^s)\|^2 \cos^2\theta_k} + \|\nabla J(x_{k+1}^s)\|}.$$

From the assumption that ∇J is Lipschitz continuous with constant L, we will have

$$\begin{aligned}\left\|\nabla J\left(x_{k+1}^s\right)\right\| &\le \left\|\nabla J\left(x_{k+1}^s\right) - \nabla J\left(\hat{x}_k^s\right)\right\| + \left\|\nabla J\left(\hat{x}_k^s\right)\right\| \\ &\le (1+\alpha_k L)\left\|\nabla J\left(\hat{x}_k^s\right)\right\|,\end{aligned}$$

and $1 \ge \cos\theta_k \ge c_3 > 0$ ($c_3 = 1$ for steepest descent algorithm) to satisfy the angle condition required for the global convergence of a line search method. Therefore

$$\begin{aligned}&\frac{(c\|\nabla J(\hat{x}_k^s)\|^2 \cos^2\theta_k)/L}{\sqrt{\|\nabla J(x_{k+1}^s)\|^2 + c\|\nabla J(\hat{x}_k^s)\|^2 \cos^2\theta_k} + \|\nabla J(x_{k+1}^s)\|} \\ &\quad\ge \frac{(c\|\nabla J(\hat{x}_k^s)\|^2 c_3^2)/L}{(\sqrt{(1+\alpha_k L)^2 + c})\|\nabla J(\hat{x}_k^s)\| + (1+\alpha_k L)\|\nabla J(\hat{x}_k^s)\|} \\ &\quad= \frac{cc_3^2\|\nabla J(\hat{x}_k^s)\|}{L(\sqrt{(1+\alpha_k L)^2 + c} + (1+\alpha_k L))}.\end{aligned}$$

Thus, let

$$b_k = \frac{cc_3^2}{L(\sqrt{(1+\alpha_k L)^2 + c} + (1+\alpha_k L))},$$

and if in finite time δ_k, the state x enters the region given by

$$\left\| x - x_{k+1}^s \right\| < b_k \left\| \nabla J\left(\hat{x}_k^s\right) \right\|, \tag{5.20}$$

then we let $\hat{x}_{k+1}^s = x(t_k + \delta_k)$ and therefore

$$\|e_{k+1}\| = \left\| \hat{x}_{k+1}^s - x_{k+1}^s \right\| \le b_k \left\| \nabla J\left(\hat{x}_k^s\right) \right\|,$$

which will satisfy the inequality (4.17).

Since we know

$$\begin{aligned} \left\| x - x_{k+1}^s \right\| &\le \left\| T^{-1}(z) - T^{-1}\left(z_{k+1}^s\right) \right\| \\ &\le L_T \left\| z - z_{k+1}^s \right\| \\ &\le L_T \left(\|z - \bar{r}_k\| + \left\| \bar{r}_k - z_{k+1}^s \right\| \right), \end{aligned}$$

then from (5.6), (5.12) and (5.20), we can solve inequality (5.21) to estimate the regulation time δ_k:

$$|e_z| + k_{\sum}\phi_\mu\left(t, |e_z|\right) + \left\| \bar{r}_k - z_{k+1}^s \right\| < \frac{b_k}{L_T} \left\| \nabla J\left(\hat{x}_k^s\right) \right\|. \tag{5.21}$$

We first consider the case when $d_2 > k_u$ in (5.10). Since $e_z(t) = e_z(0)e^{-k_u t}$, we have

$$\begin{aligned} &|e_z| + k_{\sum}\phi_\mu\left(t, |e_z|\right) + \left\| \bar{r}_k - z_{k+1}^s \right\| \\ &\quad = \left|e_z(0)\right| e^{-k_u t} + \left\| \bar{r}_k - z_{k+1}^s \right\| \\ &\qquad + k_{\sum}\left(d_1 e^{-d_2 t} \left\| \mu(0) \right\| + \frac{d_1 |e_z(0)|}{d_2 - k_u}\left(e^{-k_u t} - e^{-d_2 t}\right) \right) \\ &\quad < \left|e_z(0)\right| e^{-k_u t} + \left\| \bar{r}_k - z_{k+1}^s \right\| + k_{\sum}\left(d_1 \left\| \mu(0) \right\| e^{-k_u t} + \frac{d_1 |e_z(0)|}{d_2 - k_u} e^{-k_u t} \right) \\ &\quad = k_s e^{-k_u t} + \left\| \bar{r}_k - z_{k+1}^s \right\|, \end{aligned}$$

where

$$k_s = \left(\left|e_z(0)\right| + k_{\sum}\left(d_1 \left\| \mu(0) \right\| + \frac{d_1 |e_z(0)|}{d_2 - k_u} \right) \right).$$

Similarly, for the case when $d_2 < k_u$, we will have

$$|e_z| + k_{\sum}\phi_\mu\left(t, |e_z|\right) + \left\| \bar{r}_k - z_{k+1}^s \right\| < k_s e^{-d_2 t} + \left\| \bar{r}_k - z_{k+1}^s \right\|$$

where

$$k_s = \left(\left| e_z(0) \right| + k_{\sum} \left(d_1 \left\| \mu(0) \right\| + \frac{d_1 |e_z(0)|}{k_u - d_2} \right) \right).$$

So if $t = \bar{\delta}_k = m_k T_r$ as in (5.18) and (5.19), and considering (5.13), then it is clear that we will satisfy inequality (5.21), and therefore x will enter the neighborhood (4.17) in $\bar{\delta}_k$ time. □

Remark 5.2.3 The estimated bound $b_k \|\nabla J(\hat{x}_k^s)\|$ in (5.20) is proportional to the last gradient measurement and can be computed at the beginning of the current regulation step. Moreover, note that when $\hat{x}_k^s$ is away from the minimizer of the performance function, we will expect the gradient to be large and therefore the error bound (5.20) to be large as well. This initially looser bound will potentially help reduce the burden on the controller when the state is still far from the minimizer. Also since the step length α_k generally converges to zero as $k \to \infty$, then we have

$$b_k \to \frac{c c_3^2}{L(\sqrt{1+c}+1)}.$$

Remark 5.2.4 Since the desired set point x_{k+1}^s, b_k and $r_k(t)$ are bounded, then we will have $e_z(0)$, $\|\mu(0)\|$ bounded and therefore k_s is bounded too. Thus from (5.19), the regulation time upper bound $\bar{\delta}_k$ will be finite if the gradient $\|\nabla J(\hat{x}_k^s)\|$ is not zero. That is, this upper bound is inversely proportional to the gradient magnitude $\|\nabla J(\hat{x}_k^s)\|$. Thus, the estimation $\bar{\delta}_k$ replaces the empirical tuning of the wait time in [7], which also prevents the extremum seeking from getting stuck.

Remark 5.2.5 (Guidelines for Choosing Control Parameters) From the above analysis, we can establish some guidelines for choosing control parameters. If it is desired to one wants to decrease the regulation time δ_k, then it would be necessary to choose relatively small constants $k_1, \dots, k_{n-1}$, large control gain k_u, small d_1, large d_2, and a small period T_r for the reference signal $r_k(t)$.

5.2.3 *Algorithm and Convergence*

Even though we can estimate the upper bound $\bar{\delta}_k$, we would rather use inequality (5.20) in the ESC scheme to stop regulation, since it is easier to compute than $\bar{\delta}_k$ and less conservative. Now, we present the NOESC scheme for state feedback linearizable systems based on line search optimization method and the asymptotic state regulator designed via output tracking.

5.2.3.1 Line Search and Output-Tracking Based NOESC for State Feedback Linearizable Systems

Step 0 Given x_0^s, set $t_0 = 0$, $x_0^s = \hat{x}_0^s = x(t_0)$, and $k = 0$. Measure $J(x(t_0))$ and $\nabla J(x(t_0))$.

Step 1 Use a line search method with global convergence to produce

$$x^s_{k+1} = x(t_k) + \alpha_k p_k, \tag{5.22}$$

where α_k is the step length, and p_k is the search direction.

Step 2 Construct a reference signal $r_k(t)$ to satisfy condition (5.11), such as in (5.14).

Step 3 Apply the state regulator in (5.5).

Step 4 If the state satisfies inequality (5.20), then stop the current regulation and record the current time to be t_{k+1}. Set $\delta_k = t_{k+1} - t_k$ and $\hat{x}^s_{k+1} = x(t_{k+1})$. Measure $J(x(t_{k+1}))$ and $\nabla J(x(t_{k+1}))$.

Step 5 Set $k \leftarrow k+1$. Go to step 1.

We now have the following convergence theorem.

Theorem 5.2.6 *Consider the nonlinear system* (5.1) *with performance function* (5.2). *Suppose the system* (5.1) *is state feedback linearizable on D, with T the diffeomorphism that transforms the system to normal form* (5.3), *and* $\| \frac{\partial T^{-1}}{\partial x} \|$ *is bounded on D. Also, suppose the performance function* (5.2) *satisfies Assumptions* 3.2.1–3.2.2 *and is continuously differentiable, and ∇J is Lipschitz continuous with constant L.*

If the line search and output-tracking based ESC algorithm shown above is applied, where a maximal allowed regulation time $\bar{\delta}_k = m_k T_r$ can be computed using (5.19), *then the ESC system will be uniformly ultimately bounded* [2] (*i.e.*, *the closed-loop system is stable and the state x will globally asymptotically converge to a neighborhood of the global minimizer x^* of $J(x)$*).

Proof Now at time $t = t_k$, we have the current state $x = x(t_k) = \hat{x}^s_k$, and we obtain x^s_{k+1} as in (5.22). We apply controller (5.5) to perform output tracking of reference $r_k(t)$ and therefore regulate the state into a neighborhood of x^s_{k+1} in finite time $\bar{\delta}_k$. Thus we know we can enter the desired robust region (5.20) in a finite time, and ensure the state trajectory interpolates between the descent sequence $\{\hat{x}^s_k\}$ every $\bar{\delta}_k$ time.

According to Assumption 3.2.2, we suppose that the unknown global minimizer is an isolated equilibrium point. Therefore, let $e = x - x^*$, and we choose the Lyapunov candidate

$$\begin{aligned} V(e,k) &= J\big(e(t_k) + x^*\big) - J\big(x^*\big) \\ &= J\big(x(t_k)\big) - J\big(x^*\big), \end{aligned}$$

which is positive whenever $e \neq 0$ and zero for $e = 0$. And we know that the controller will ensure that the state x crosses $\hat{x}^s_k$ and $\hat{x}^s_{k+1}$ (where $\hat{x}^s_{k+1} = \hat{x}^s_k + \alpha_k p_k + e_{k+1}$) at time $t = t_k$ and $t_{k+1} = t_k + \bar{\delta}_k$, respectively. Then, according to Theorem 4.2.1, and assuming $\hat{x}^s_{k+1} \neq x^*$, and $\bar{\delta}_k = m_k T_r$ satisfies (5.19), we will have $\Delta V = V(e,k+1) - V(e,k) = J(x(t_{k+1})) - J(x(t_k)) = J(\hat{x}^s_{k+1}) - J(\hat{x}^s_k) < 0$. Thus, we can conclude the closed-loop system is stable and x will converge to a neighborhood of x^*. That is, the ESC system is uniformly ultimately bounded. And from equations (5.18) and (5.19), we can obtain the size of the neighborhood,

$$\left\|\nabla J\left(\hat{x}_k^s\right)\right\| \leq \frac{L_T k_s}{b_k e^{\min(k_u, d_2)\bar{\delta}_k}}, \tag{5.23}$$

which can be make arbitrary small given large enough k_u, d_2 or $\bar{\delta}_k$. □

Remark 5.2.7 The change of coordinate T is for the sake of designing the asymptotic state regulator (5.5) based on the feedback linearized system (5.3) output tracking a reference signal, but the desired set point is still generated via the optimization algorithm in x coordinates.

Remark 5.2.8 The measurement of gradient is required because a line search method is used, but we can relax such requirement by estimating the gradient given it is only needed every δ_k time. Also, we certainly can combine a non-gradient optimization algorithm with the asymptotic state regulator to form non-gradient NOESC [8]. The robustness result of trust region methods can be found in Sect. 4.2.2, where now we need to transform that result into a region similar to that defined via the bound (5.20): first, we need to estimate the gradient at x_{k+1}^s and therefore obtain the size of the robust region, then we can shift the regulation set point x_{k+1}^s to the center of the robust region.

Remark 5.2.9 One could argue that given knowledge of x and measurements of y, why not simply estimate J and then pursue a robust nonlinear control design. First, even if the performance function is known perfectly, we need to compute the optimal set point by finding the root of its gradient, which may be difficult as well. Mathematically, the root finding and optimization problem are equivalently difficult to solve. Second, the trust region method (a class of optimization algorithms) does in fact use an estimated model of J in a trusted region to perform optimization step by step.

Remark 5.2.10 A stopping criterion like $\|\nabla J(x(t_{k+1}))\| < \varepsilon_0$ can be used to terminate the extremum seeking loop in finite iterations, where ε_0 is a predefined small positive constant. It is equivalent to having an upper limit on the regulation time δ_k. In the case when gradient information is not available, there are other stopping criteria only based on the difference of function values [5].

Remark 5.2.11 An important point to note is that the NOESC framework does not explicitly separate the plant dynamics from the extremum seeking loop, and therefore does not assume time scale separation between them, as PESC and SMESC typically do. At the same time, a time separation between the optimizer and the control loop does emerge in practice, due to the structure of the scheme: as the optimizer block provides the sequence of set points x_s^k, there is a regulation time the controller needs in order to move the state between the set points. This regulation time is not fixed at each step of the optimization procedure, but we know it is finite, and bounded by $\bar{\delta}_k$.

Example 5.2 Consider a second order nonlinear system

$$\begin{aligned}\dot{x}_1 &= -x_1 + x_2,\\ \dot{x}_2 &= x_1 x_2 + u,\end{aligned} \tag{5.24}$$

with the Rosenbrock performance function

$$y = J(x) = 100\big(x_2 - x_1^2\big)^2 + (1 - x_2)^2. \tag{5.25}$$

The performance function (5.25) has its minimizer at $x^* = [1, 1]^\top$, and the minimum is $J(1, 1) = 0$, which is a stabilizable equilibrium by letting $u(t) = -1$. The explicit form of the performance function and its minimizer are both assumed unknown to the designer.

Let $z_1 = x_1$, $z_2 = -x_1 + x_2$. The transformed system in the new coordinates (z_1, z_2) is

$$\begin{aligned}\dot{z}_1 &= z_2,\\ \dot{z}_2 &= -(-x_1 + x_2) + x_1 x_2 + u.\end{aligned}$$

In the NOESC scheme, we only need to measure the function value and its gradient value every δ_k time. Then, at iteration $k + 1$, we use the line search method to obtain the set point

$$x^s_{k+1} = x(t_k) - \alpha_k \nabla J\big(x(t_k)\big).$$

We then compute $z^s_{k+1} = T(x^s_{k+1})$, that is

$$\begin{aligned}z^s_{k+1,1} &= x^s_{k+1,1},\\ z^s_{k+1,2} &= -x^s_{k+1,1} + x^s_{k+1,2}.\end{aligned}$$

The bounded reference signal is chosen to have period T_r as

$$r_k(t) = a_1 \sin(\omega_1 t) + a_2,$$

where $\omega_1 = 2\pi/T_r$, $a_1 = z^s_{k+1,2}/\omega_1$ and $a_2 = z^s_{k+1,1}$. It is easy to verify that this design satisfies conditions (5.11). The error manifold is defined as $e_z = k_1(z_1 - r_k) + z_2 - \dot{r}_k$. Then, the controller (5.5) becomes

$$u = -k_1(z_2 - \dot{r}_k) + \ddot{r}_k + (-x_1 + x_2) - x_1 x_2 - k_u e_z. \tag{5.26}$$

We now implement the NOESC scheme in Sect. 5.2.3, where the initial condition is $x^s_0 = [-1.9, 0]^\top$, $t_0 = 0$ and line search method with $p_k = -\nabla J(x(t_k))$, i.e.,

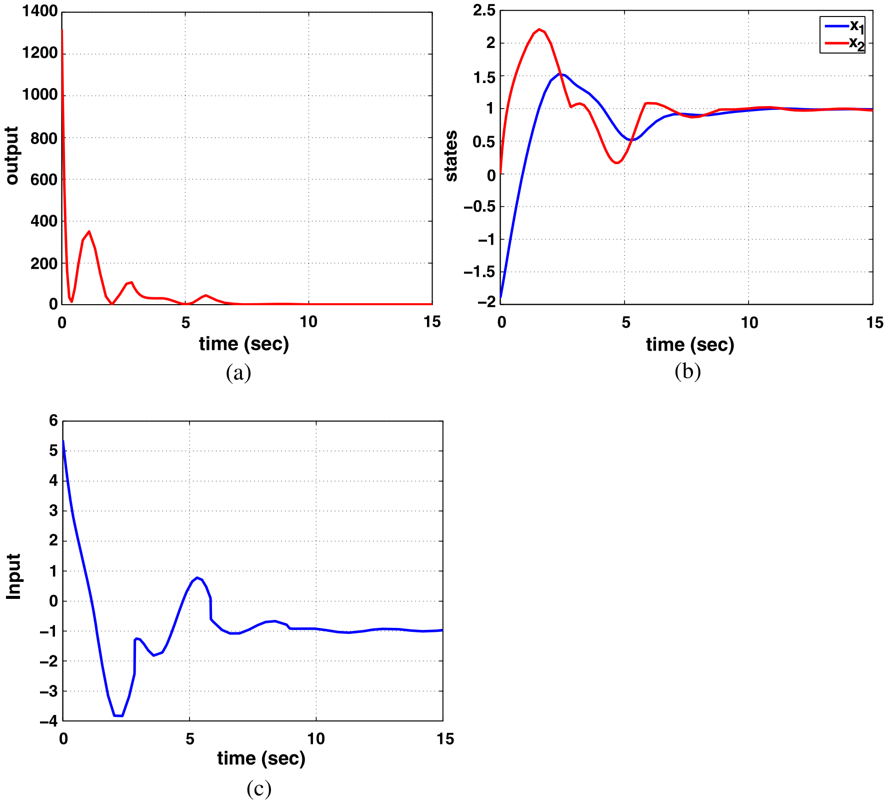


Fig. 5.1 Line search and output-tracking based NOESC, $k_1 = 2$, $k_u = 2$, $T_r = 3$, $\alpha_k = 0.0012$: (**a**) performance output; (**b**) state; (**c**) control input

steepest descent algorithm. The designer chooses step length α_k, gains k_1, k_u, and estimates the b_k in (5.20). The simulation results can be found in Fig. 5.1, where the performance output (5.25) (Fig. 5.1(a)) approaches its minimum at $J(1, 1) = 0$ and the state (Fig. 5.1(b)) accordingly converges to the minimizer $[1, 1]^\top$. The control input can be seen in Fig. 5.1(c). The steepest descent algorithm produces a set point sequence $\{x_k^s\}$ as commands for the state regulation. The trajectory between x_k^s and x_{k+1}^s is shaped by the dynamic system (5.24) and the state regulator (5.26). This can be viewed clearly in Fig. 5.2, where the blue circles represent the $\{x_k^s\}$ and the red dashed lines represent the state trajectory.

It is also worth noticing that the control only regulates the state to the neighborhood of the set point, as seen from Fig. 5.2. That is, the control's action results in the state trajectory interpolating between the points in the sequence $\{\hat{x}_k^s\}$. Moreover, one can choose a large control gain k_u, and a small period of the reference signal T_r to accelerate the extremum seeking loop. Finally, we note that the control (5.26) converges to -1, which stabilizes the system at minimizer $[1, 1]^\top$. Therefore the steady

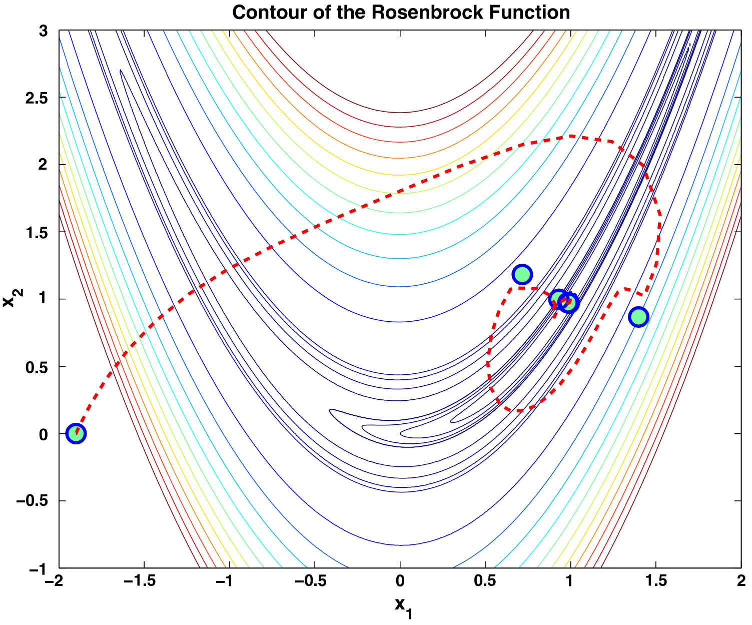


Fig. 5.2 Line search and output tracking based NOESC, $k_1 = 2$, $k_u = 2$, $T_r = 3$, $\alpha_k = 0.0012$: phase portrait

state output does not oscillate by using the asymptotic state regulator, in contrast to the finite-time state regulator of Example 4.2.

5.3 Robust Design for Input Disturbance

In this and the following sections, we will see the advantage of the asymptotic state regulator design, where we trade off the finite-time state regulation to achieve robust state regulation, i.e., robustly regulating x to x^s_{k+1}, or equivalently, driving the error e_z in (5.4) to 0. Now, consider an input disturbance satisfying the matching condition

$$\dot{e}_z = \chi(z) + f(z) + g(z)\big(u + \varDelta(t,z)\big), \tag{5.27}$$

where $\varDelta(t,z)$ is the unknown input disturbance (we postulate it as a function of z for notation convenience; mathematically it is equivalent to having it as a function of x). Nonlinear damping is used to overcome the disturbance [6]. That is, the new state regulator for robust extremum seeking control is

$$u = \frac{-\chi(z) - f(z) - k_u e_z}{g(z)} + u_s, \quad k_u > 0, \tag{5.28}$$

where u_s is the stabilizing term designed based on the properties of the disturbance.

5.3.1 Bounded Input Disturbance

First, we assume $|\Delta(t,z)| \leq \beta$ represents some bounded disturbance with $\beta > 0$ a known constant. We can first choose

$$u_s = -\beta \,\mathrm{sgn}\big(e_z g(z)\big), \tag{5.29}$$

where $\mathrm{sgn}(\cdot)$ is the signum function , defined as

$$\mathrm{sgn}(y) = \begin{cases} 1, & y \geq 0, \\ -1, & y < 0. \end{cases} \tag{5.30}$$

Now consider the Lyapunov candidate $V = (1/2)e_z^2$, and we find that $\dot{V} = e_z \dot{e}_z = -2kV + e_z g(z)(u_s + \Delta(t,z))$. Then we have

$$\dot{V} \leq -2kV - \big|e_z g(z)\big|\beta + \big|e_z g(z)\big|\big|\Delta(t,z)\big| \leq -2kV,$$

from which we conclude that the error e_z asymptotically converges to zero using the nonlinear damping term (5.29).

Since u_s in (5.29) is discontinuous, one may not be able to guarantee existence and uniqueness of the solutions of the plant's differential equation. Moreover, from a practical point of view, a discontinuous control signal may be unduly harsh on actuators, and it may also excite unmodeled high frequency plant dynamics. To avoid these issues, it is possible to choose instead a smoothed approximation of the nonlinear damping term (5.29), such as

$$u_s = -\beta \frac{e_z g(z)}{|e_z g(z)| + c_s}, \tag{5.31}$$

with $c_s > 0$ a small constant. Then instead of having the origin of e_z rendered asymptotically stable, we will only be able to conclude that it is uniformly ultimately bounded, a more practical and realistic result. To show this, consider again the Lyapunov candidate $V = (1/2)e_z^2$, where now we use the smoothed approximation (5.31). Then we find

$$\begin{aligned} \dot{V} &\leq -2kV + e_z g(z)\left(-\beta \frac{e_z g(z)}{|e_z g(z)| + c} + \Delta(t,z)\frac{|e_z g(z)| + c}{|e_z g(z)| + c}\right) \\ &\leq -2kV + \frac{\beta}{|e_z g(z)| + c}\left(-\big|e_z g(z)\big|^2 + \big|e_z g(z)\big|^2 + c\big|e_z g(z)\big|\right) \\ &= -2kV + \frac{c\beta |e_z g(z)|}{|e_z g(z)| + c} \\ &\leq -2kV + c\beta. \end{aligned} \tag{5.32}$$

We find $\dot{V} < 0$ whenever $V > c\beta/2k$, or equivalently when $|e_z| > \sqrt{c\beta/k}$.

The size of the neighborhood can be made arbitrarily small given large enough k_u and small enough c_s. There are other approximations of the signum function available and will arrive at similar results.

5.3.2 Unbounded Input Disturbance

Here, we assume $|\Delta(t,z)| \le \beta\psi(z)$ with β an unknown constant and $\psi : \mathbb{R}^n \to \mathbb{R}$ is a known nonnegative function. It is only assumed that ψ is bounded for any bounded $z \in \mathbb{R}^n$, in this case, we have the disturbance Δ may grow unbounded if $z \to \infty$. Then, we consider the stabilizing term

$$u_s = -\eta e_z g(z)\psi^2(z), \quad \eta > 0. \tag{5.33}$$

The time derivative of the Lyapunov candidate $V = \frac{1}{2}e_z^2$ becomes

$$\begin{aligned}\dot{V} &= -2kV + e_z g(z)\big(-\eta e_z g(z)\psi^2(z) + \Delta(t,z)\big)\\ &\le -2kV - \eta\big|e_z g(z)\big|^2\psi^2(z) + \beta\big|e_z g(z)\big|\psi(z) - 2kV + \frac{\beta^2}{4\eta}.\end{aligned}$$

We find $\dot{V} < 0$ whenever $V > \beta^2/(8k\eta)$, or equivalently when $|e_z| > \beta/\sqrt{4k\eta}$. This means that the origin of e_z is uniformly ultimately bounded and hence the closed-loop system is stable. As before, the size of the neighborhood of e_z can be made arbitrarily small given large enough k_u and η.

Now we will examine the implications of uniform ultimate boundedness [2] of e_z, as opposed to asymptotic convergence to zero. Note that the convergence of the numerical optimization based ESC relies on the controller to drive the state to the desired set point x_{k+1}^s or within the required neighborhood in finite time. When e_z asymptotically converges to zero (such as when using the term (5.29) in the bounded input disturbance case), we know from Theorem 5.2.2 that there exists a $\bar{\delta}_k$ such that the inequality (5.21) is valid for some $t \le \bar{\delta}_k$.

Consider instead the case when e_z is uniformly ultimately bounded given an input disturbance. For example, in the case of a bounded disturbance with smooth damping term (5.31), one obtains from (5.32) that

$$\big|e_z(t)\big| \le \sqrt{\frac{c\beta}{k} + \left(2V(0) - \frac{c\beta}{k}\right)e^{-2kt}}.$$

Therefore, we have $\lim_{t\to\infty} |e_z(t)| \le \sqrt{c\beta/k}$. Denote $\bar{e}_z = \max\{\sqrt{2V(0)}, \sqrt{c\beta/k}\}$. Then, from (5.6) and (5.10),

$$\begin{aligned}\phi_\mu\big(t,|e_z|\big) &= d_1 e^{-d_2 t}\big\|\mu(0)\big\| + \int_0^t d_1 e^{-d_2(t-\tau)}\big|e_z(\tau)\big|\,d\tau\\ &\le d_1 e^{-d_2 t}\big\|\mu(0)\big\| + d_1\bar{e}_z e^{-d_2 t}\int_0^t e^{d_2\tau}\,d\tau\\ &\le d_1 e^{-d_2 t}\big\|\mu(0)\big\| + d_1\bar{e}_z\big(1 - e^{-d_2 t}\big)/d_2.\end{aligned}$$

Then, recalling (5.13), for $t = \bar{\delta}_k = m_k T_r$ we have

$$\begin{aligned}&\big(|e_z| + k_\sum \phi_\mu\big(t, |e_z|\big) + \big\|\bar{r}_k - z^s_{k+1}\big\|\big)_{t=\bar{\delta}_k}\\&\quad\le \bar{e}_z + k_\sum\big(d_1 e^{-d_2\bar{\delta}_k}\big\|\mu(0)\big\| + d_1\bar{e}_z\big(1 - e^{-d_2\bar{\delta}_k}\big)/d_2\big).\end{aligned}$$

The system will still be uniformly ultimately bounded, but the size of the neighborhood will be larger than in (5.23) and can be given as

$$\big\|\nabla J\big(\hat{x}^s_k\big)\big\| \le \frac{L_T}{b_k}\big[\bar{e}_z + k_\sum\big(d_1 e^{-d_2\bar{\delta}_k}\big\|\mu(0)\big\| + d_1\bar{e}_z\big(1 - e^{-d_2\bar{\delta}_k}\big)/d_2\big)\big], \tag{5.34}$$

which can be made arbitrary small. In general, one wants to have a small $\bar{e}_z$ (choose large k_u, small β for the bounded disturbance case), and choose large d_2, $\bar{\delta}_k$ and small c, d_1, $k_1, k_2, \dots, k_{n-1}$ to have a small size neighborhood.

Example 5.3 Simulation configurations are the same as in Example 5.2. Let the bounded input disturbance be

$$\Delta(t, z) = 2\,\mathrm{rand}(t) \tag{5.35}$$

where rand(t) is uniformly distributed noise in the range $[-1, 1]$ with amplitude 1. Also consider an unbounded disturbance given by

$$\Delta(t, z) = \frac{1}{|z_1| + \mathrm{rand}(t)} + 3\big(\cos(t) + 1\big)z_2, \tag{5.36}$$

which can be bounded by $|\Delta(t, z)| \le \frac{1}{|z_1|+1} + 6z(2)$.

The simulation results of the nominal controller (5.5) given input disturbances are shown in Fig. 5.3, where we have the state converge to a neighborhood of x^* due to the bounded disturbance, and the system becomes unstable given the unbounded disturbance.

Now, the robust extremum seeking controller (5.28) is introduced to deal with the bounded input disturbance (5.35). The simulation results for stabilizing term (5.29) can be found in Fig. 5.4. Even though the stabilizing term with signum function achieves good performance, it produces chattering in the control input as seen in Fig. 5.4(c). As seen in Fig. 5.5, the approximated version (5.31) implements a smoothed control law and achieves comparable results.

For the unbounded disturbance (5.36), the stabilizing term (5.33) is able to overcome the unbounded disturbance and achieve the extremum seeking purpose, as seen in Fig. 5.6.

5.4 Robust Design for Unknown Plant Dynamics

The state regulator (5.5) is based on exact mathematical cancelation of the nonlinear terms $f(z)$ and $g(z)$. This is generally difficult in practice for several reasons such

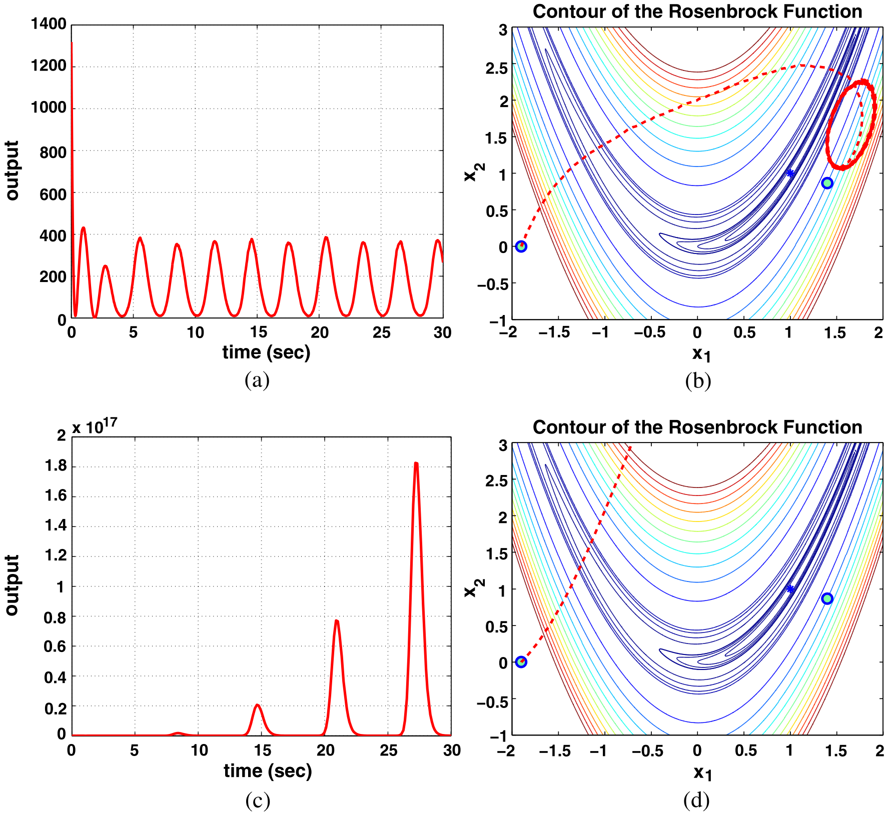


Fig. 5.3 Line search and output-tracking based NOESC, $k_1 = 2$, $k_u = 2$, $\alpha_k = 0.0012$: (**a**) performance output of bounded input disturbance; (**b**) phase portrait of bounded input disturbance; (**c**) performance output of unbounded input disturbance; (**d**) phase portrait of unbounded input disturbance

as model simplification, parameter uncertainty and computational errors. Usually, we will implement instead the feedback control law

$$u = \frac{-\chi(z) - \hat{f}(z) - k_u e_z}{\hat{g}(z)},$$

where $\hat{f}(z), \hat{g}(z)$ are approximations of $f(z)$ and $g(z)$. One method to deal with the approximation error is to treat it as an input disturbance. Then, we can design static (non-adaptive) stabilizing controllers to deal with input disturbance as seen in Sect. 5.3. Here, instead, we use approximation based adaptive control laws [6] to deal with unknown plant dynamics, where we remove the assumption of exact knowledge of plant dynamics and only assume that $g(z) \geq g_0 > 0$.

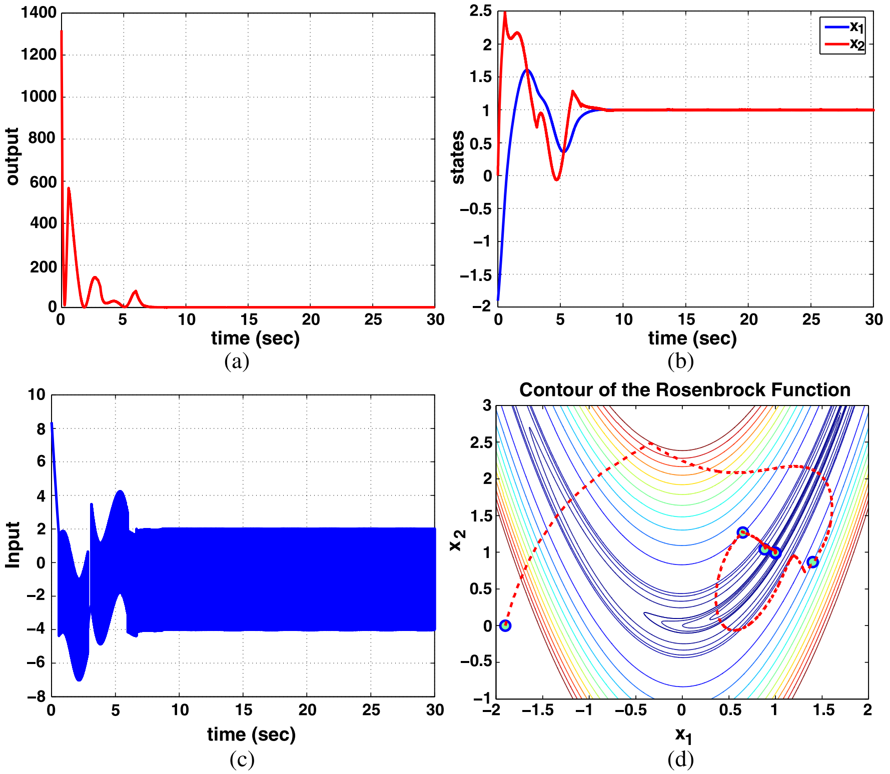


Fig. 5.4 Stabilizing controller using signum function for bounded input disturbance, $k_1 = 2$, $k_u = 2$, $\alpha_k = 0.001$, $\beta = 3$: (**a**) performance output; (**b**) state; (**c**) control input; (**d**) phase portrait

5.4.1 Indirect Adaptive Control

First we will approximate the unknown plant dynamics $f(z)$ and $g(z)$ using two function approximators, and then use them to construct an adaptive controller. Now, we assume that the function approximator will approximate the plant dynamics within a compact set $S \subset \mathbb{R}^n$. That is,

$$f(z) = \mathcal{F}_1\big(z, \theta_1^*\big) + \omega_1(z),$$
$$g(z) = \mathcal{F}_2\big(z, \theta_2^*\big) + \omega_2(z),$$

where

$$\mathcal{F}_1\big(z, \theta_1^*\big) = \theta_1^{*\top} \xi_1(z),$$
$$\mathcal{F}_2\big(z, \theta_2^*\big) = \theta_2^{*\top} \xi_2(z)$$

are function approximators using basis functions $\xi_1(z)$, $\xi_2(z)$.

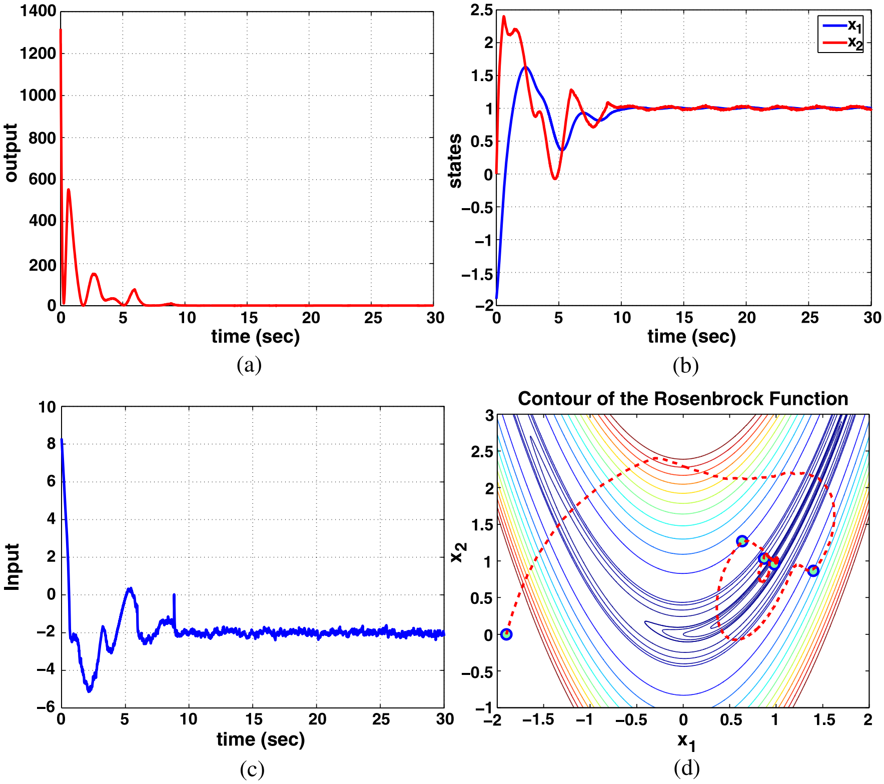


Fig. 5.5 Smoothed stabilizing controller for bounded input disturbance, $k_1 = 2$, $k_u = 2$, $\alpha_k = 0.0012$, $\beta = 3$, $c_s = 0.1$: (**a**) performance output; (**b**) state; (**c**) control input; (**d**) phase portrait

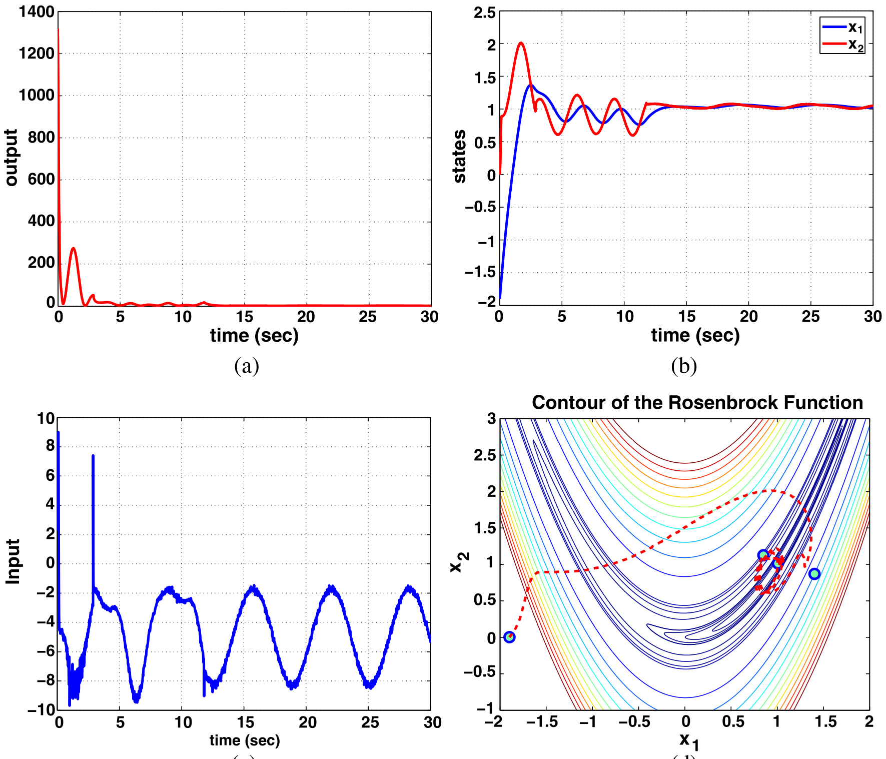
Moreover, $\theta_1^* \in \mathbb{R}^{p_1}, \theta_2^* \in \mathbb{R}^{p_2}$ are unknown optimal parameters such that for arbitrary $z \in S$ we have $|\omega_1(z)| \le W_1$ and $|\omega_2(z)| \le W_2$ for some known constants W_1 and W_2, which are the smallest possible given p_1, p_2 and S. Let $\hat{\theta}_1, \hat{\theta}_2$ be the estimates of θ_1^*, θ_2^*, with parameter error vectors $\tilde{\theta}_1 = \hat{\theta}_1 - \theta_1^*$ and $\tilde{\theta}_2 = \hat{\theta}_2 - \theta_2^*$. Now we design the indirect adaptive controller to be

$$u = \underbrace{\frac{1}{\mathcal{F}_2(z, \hat{\theta}_2)}\big(-\chi(z) - \mathcal{F}_1(z, \hat{\theta}_1) - k_u e_z\big)}_{u_{FL}} + u_s. \tag{5.37}$$

We choose the Lyapunov candidate

$$V = \frac{1}{2}e_z^2 + \frac{1}{2\gamma_1}\tilde{\theta}_1^\top \tilde{\theta}_1 + \frac{1}{2\gamma_2}\tilde{\theta}_2^\top \tilde{\theta}_2,$$

where γ_1, γ_2 are some positive constants. Note that



(a) (b)

(c) (d)

Fig. 5.6 Stabilizing controller for unbounded input disturbance, $k_1 = 1$, $k_u = 3$, $\alpha_k = 0.0012$, $\eta = 2$: (**a**) performance output; (**b**) states; (**c**) control input; (**d**) phase portrait

$$
\begin{aligned}
\dot{e}_z &= \chi(z) + f(z) + g(z)(u_{FL} + u_s) \\
&= \chi(z) + f(z) + g(z)(u_{FL} + u_s) + \mathcal{F}_2\big(z, \hat{\theta}_2\big)(u_{FL} - u_{FL}) \\
&= \big(\mathcal{F}_1\big(z, \theta_1^*\big) + \omega_1(z)\big) + \big(\mathcal{F}_2\big(z, \theta_2^*\big) + \omega_2(z)\big)(u_{FL} + u_s) \\
&\quad - \mathcal{F}_1\big(z, \hat{\theta}_1\big) - ke_z - \mathcal{F}_2\big(z, \hat{\theta}_2\big)u_{FL} \\
&= -ke_z - \mathcal{F}_1\big(z, \tilde{\theta}_1\big) - \mathcal{F}_2\big(z, \tilde{\theta}_2\big)u_{FL} + \omega_1(z) + \omega_2(z)u_{FL} + g(z)u_s.
\end{aligned}
$$

Then,

$$
\begin{aligned}
\dot{V} &= e_z\dot{e}_z + \frac{1}{\gamma_1}\tilde{\theta}_1^\top\dot{\hat{\theta}}_1 + \frac{1}{\gamma_2}\tilde{\theta}_2^\top\dot{\hat{\theta}}_2 \\
&= -ke_z^2 - \tilde{\theta}_1^\top\xi_1(z)e_z - \tilde{\theta}_2^\top\xi_2(z)u_{FL}e_z + \frac{1}{\gamma_1}\tilde{\theta}_1^\top\dot{\hat{\theta}}_1 + \frac{1}{\gamma_2}\tilde{\theta}_2^\top\dot{\hat{\theta}}_2 \\
&\quad + \omega_1(z)e_z + \omega_2(z)u_{FL}e_z + g(z)u_se_z.
\end{aligned}
$$

Choose the parameter update laws and the stabilizing term to be

$$\dot{\hat{\theta}}_1 = \gamma_1 \xi_1(z) e_z, \tag{5.38}$$

$$\dot{\hat{\theta}}_2 = \gamma_2 \xi_2(z) u_{FL} e_z, \tag{5.39}$$

$$u_s = \frac{1}{g_0}\bigl(-W_1 - W_2 |u_{FL}|\bigr)\operatorname{sgn}(e_z). \tag{5.40}$$

Now, we have

$$\begin{aligned}\dot{V} &\le -ke_z^2 + \bigl|\omega_1(z)e_z\bigr| + \bigl|\omega_2(z)u_{FL}e_z\bigr| + g(z)u_s e_z \\ &\le -ke_z^2 + W_1|e_z| + W_2|u_{FL}||e_z| + \frac{g(z)}{g_0}\bigl(-W_1 - W_2|u_{FL}|\bigr)|e_z| \\ &\le -ke_z^2.\end{aligned}$$

Therefore, we can first see that the system is stable and e_z, $\tilde{\theta}_1$, $\tilde{\theta}_2$ are bounded. Furthermore, e_z asymptotically converges to zero [6].

5.4.2 Direct Adaptive Control

As an alternative approach to the adaptive NOESC problem, instead of approximating the plant dynamics we can directly approximate the controller (5.5). In order to do this, we need two more assumptions: $0 < g_0 \le g(z) < g_1 \le \infty$ and $|\dot{g}(z)| \le B_1 < \infty$ for $z \in S$, where g_0 and B_1 are known constants.

Now let u_{FL}^* represent the nominal controller (5.5), and we assume that the function approximator will approximate it within a compact set $S \subset \mathbb{R}^n$, that is,

$$u_{FL}^* = \mathcal{F}_u\bigl(z, \theta_u^*\bigr) + \omega_u(z),$$

where

$$\mathcal{F}_u\bigl(z, \theta_u^*\bigr) = \theta_u^{*\top} \xi_u(z)$$

is a function approximator using basis function $\xi_u(z)$. Moreover, $\theta_u^* \in \mathbb{R}^{p_u}$ are unknown optimal parameters such that for arbitrary $z \in S$ we have $|\omega_u(z)| \le W_u$ for some known constant W_u, which is the smallest possible given p_u, S. Let $\hat{\theta}_u$ be the estimate of θ_u^* and $\tilde{\theta}_u = \hat{\theta}_u - \theta_u^*$. Now, we design the direct adaptive controller as

$$u = \mathcal{F}_u(z, \hat{\theta}_u) + u_s, \tag{5.41}$$

where u_s is a stabilizing term defined later. We choose the Lyapunov candidate

$$V = \frac{1}{2g(z)} e_z^2 + \frac{1}{2\gamma_u} \tilde{\theta}_u^\top \tilde{\theta}_u,$$

where γ_u is a positive constant. Note that

$$\begin{aligned}
\dot{e}_z &= \chi(z) + f(z) + g(z)u \\
&= \chi(z) + f(z) + g(z)\big(\mathcal{F}_u(z, \hat{\theta}_u) + u_s + \mathcal{F}_u\big(z, \theta_u^*\big) - \mathcal{F}_u\big(z, \theta_u^*\big)\big) \\
&= \chi(z) + f(z) + g(z)\big(\tilde{\theta}_u^\top \xi_u(z) + u_s + u_{FL}^* - \omega_u(z)\big) \\
&= -ke_z + g(z)\big(\tilde{\theta}_u^\top \xi_u(z) + u_s - \omega_u(z)\big).
\end{aligned}$$

Then,

$$\begin{aligned}
\dot{V} &= \frac{1}{g(z)} e_z \dot{e}_z - \frac{\dot{g}(z) e_z^2}{2g(z)^2} + \frac{1}{\gamma_u} \tilde{\theta}_u^\top \dot{\hat{\theta}}_u \\
&= -\frac{k}{g(z)} e_z^2 + \tilde{\theta}_u^\top \xi_u(z) e_z + \big(u_s - \omega_u(z)\big) e_z - \frac{\dot{g}(z) e_z^2}{2g(z)^2} + \frac{1}{\gamma_u} \tilde{\theta}_u^\top \dot{\hat{\theta}}_u.
\end{aligned}$$

Choose the parameter update law and stabilizing term to be

$$\dot{\hat{\theta}}_u = -\gamma_u \xi_u(z) e_z \tag{5.42}$$

$$u_s = -\left(W_u + \frac{B_1}{2g_0^2}|e_z|\right) \operatorname{sgn}(e_z). \tag{5.43}$$

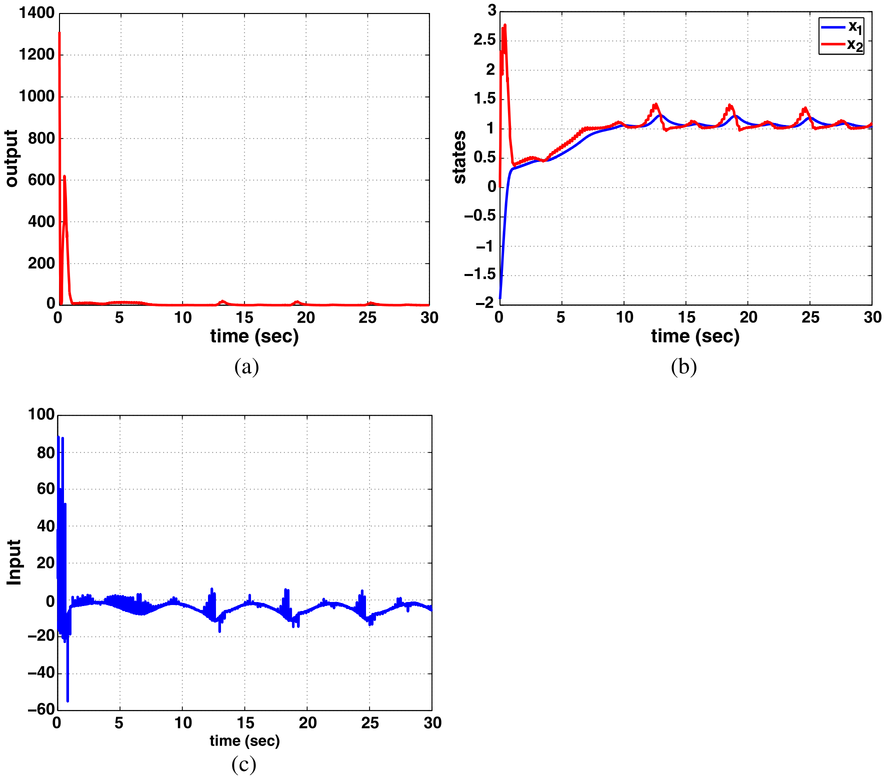
Now, we have

$$\begin{aligned}
\dot{V} &\leq -\frac{k}{g(z)} e_z^2 + \big|w_u(z) e_z\big| + \left|\frac{\dot{g}(z) e_z^2}{2g(z)^2}\right| + u_s e_z \\
&\leq -\frac{k}{g(z)} e_z^2 + W_u |e_z| + \frac{B}{2g_0^2}|e_z|^2 - \left(W_u + \frac{B}{2g_0^2}|e_z|\right)|e_z| \leq -\frac{k}{g(z)} e_z^2 \leq -\frac{k}{g_1} e_z^2.
\end{aligned}$$

Therefore, we can first see that the system is stable and e_z, $\tilde{\theta}_u$ are bounded. We have also that e_z asymptotically converges to zero as in [6].

Remark 5.4.1 We can relax the assumption of knowing W_1, W_2 and W_u, which can be estimated online as well. Also, note that both stabilizing terms (5.40) and (5.43) use the signum function. We can choose instead a continuous term to approximate the signum function, similar to (5.31). In this case, we will see that the error e_z is uniformly ultimately bounded instead of asymptotically convergent to zero. Therefore, the closed-loop system will be stable and the state x will asymptotically converge to an arbitrarily small neighborhood of the global minimizer x^*.

Example 5.4 Same simulation configurations as in Sect. 5.2 are used here. Simulation results for indirect adaptive control can be found in Figs. 5.7 and 5.8, whereas the direct adaptive control case appears in Figs. 5.9 and 5.10. In both cases, we use



(a)

(b)

(c)

Fig. 5.7 Indirect adaptive control for unknown plant dynamics with unknown unbounded input disturbance, $k_1 = 2$, $k_u = 2$, $p_1 = p_2 = 3$, $\alpha_k = 0.0007$, $g_0 = 0.9$, $W_1 = 10$, $W_2 = 10$, $\gamma_1 = 2$, $\gamma_2 = 2$: (**a**) performance output; (**b**) state; (**c**) control input

a saturation function to approximate the signum function in the stabilizing terms. The saturation function is defined as

$$\operatorname{sat}(y) = \begin{cases} 1, & y > 1, \\ y, & -1 \leq y \leq 1, \\ -1, & y < -1. \end{cases} \tag{5.44}$$

Thus, in the stabilizing terms (5.40) and (5.43), we can use the approximation

$$\operatorname{sgn}(e_z) \approx \operatorname{sat}(e_z/\varepsilon),$$

where ε is a design parameter that controls the width of the transition from negative to positive.

We can see that both adaptive controllers not only deal with the unknown plant dynamics, but also the unknown input disturbance. In the indirect adaptive control case, the effect of input disturbance is taken into account in $f(z)$, and the function approximator will try to approximate $f(z) + g(z)\Delta(z)$ together. Similarly, in the direct adaptive control case, the function approximator will approximate the robust

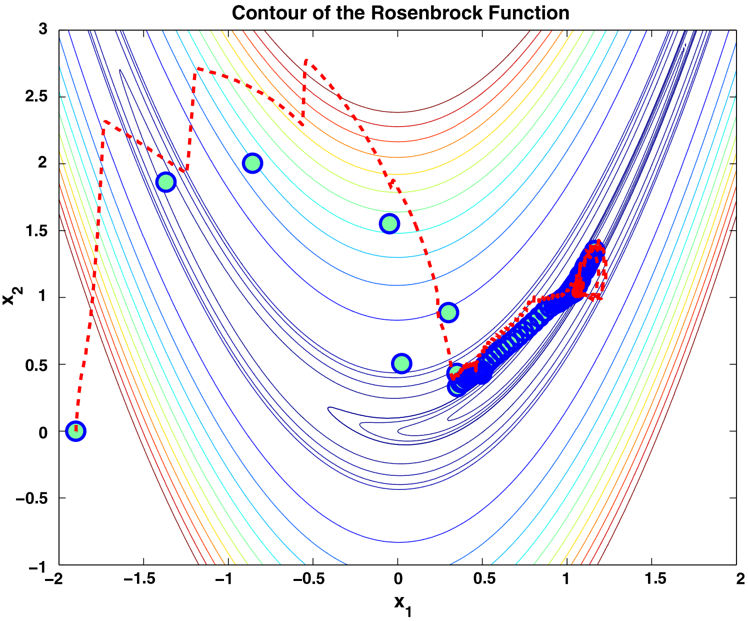


Fig. 5.8 Indirect adaptive control for unknown plant dynamics with unknown unbounded input disturbance, $k_1 = 2$, $k_u = 2$, $p_1 = p_2 = 3$, $\alpha_k = 0.0007$, $g_0 = 0.9$, $W_1 = 10$, $W_2 = 10$, $\gamma_1 = 2$, $\gamma_2 = 2$: phase portrait

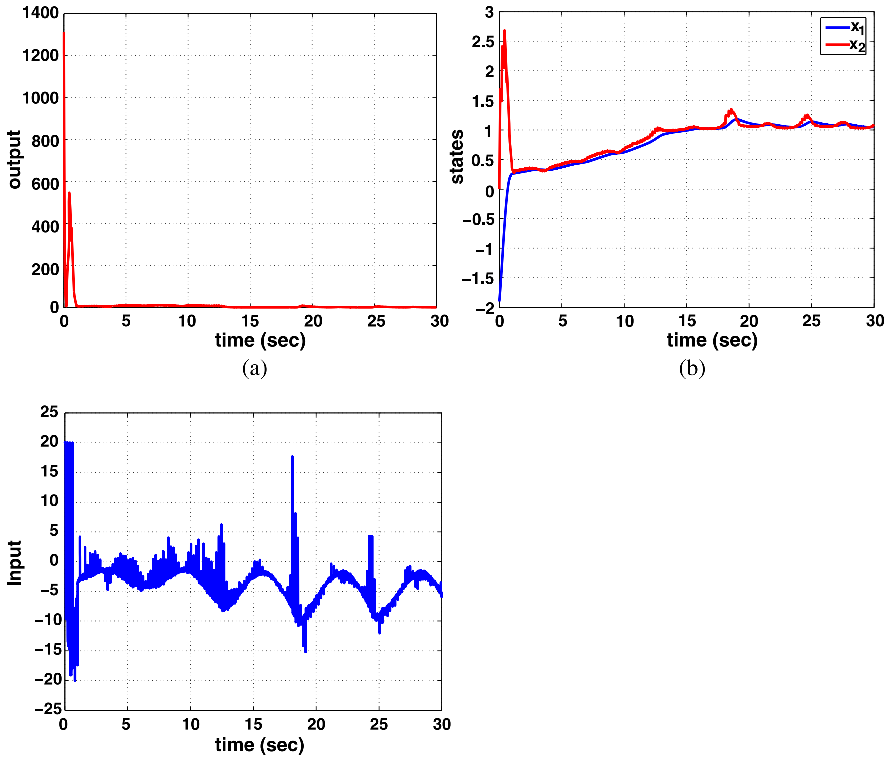
controller such as (5.28) to deal with the input disturbance implicitly. Therefore, we have seen that advanced control techniques can be incorporated in the robust design of NOESC.

5.5 Conclusions

In this chapter, inspired by the standard output-tracking problem [6], we propose a new design of state regulator for state feedback linearizable systems, and then extend this design to robust NOESC. The robustness of the numerical optimization algorithms (in particular, line search algorithm) enables the applicability of the asymptotic state regulator, which leads to the robust NOESC design for input disturbances and unknown plant dynamics, using nonlinear damping and function approximation based adaptive control techniques, respectively.

We have assumed in general that the system is globally feedback linearizable by letting $D = \mathbb{R}^n$. This is a strong assumption, which can be relaxed by considering a nonlinear system that is only feedback linearizable on $D \subset \mathbb{R}^n$. Furthermore, we may want to put additional algebraic constraints on the performance function $J(x)$. For example, we could require the state to satisfy certain inequalities constraints of the form

$$a_i \leq x_i \leq b_i, \quad i \in \{1, \ldots, n\},$$



(a) (b) (c)

Fig. 5.9 Direct adaptive control for unknown plant dynamics with unknown unbounded input disturbance, $k_1 = 2$, $k_u = 2$, $p_u = 3$, $\alpha_k = 0.004$, $g_0 = 0.9$, $B_1 = 0$, $W_u = 20$, $\gamma_u = 5$: (**a**) performance output; (**b**) state; (**c**) control input

which can be simplified as

$$\min_x J(x) \quad \text{subject to } x \in E \subset \mathbb{R}^n,$$

with E appropriately defined.

In Sect. 5.4 we use function approximators to approximate either the dynamics of the plant (indirect adaptive control case) or the nominal controller (direct adaptive control). In both of these cases, the approximation is only valid on a subset S of $\mathbb{R}^n$. The controller design has not yet been addressed to guarantee that the state will not exit the subset S (although this can be achieved using, e.g., using high-gain bounding control terms or via analysis techniques similar to those in [6]). Thus, ESC with state constraints will be an important future research topic.

To this end, some basic ideas can be formulated. Let $S_x \subseteq D \cap E \cap T^{-1}(S)$. First, we would need a constrained optimization algorithm (for example, penalty methods, or barrier methods [5]) to generate the set point sequence x^s_{k+1} inside S_x. Then we can use Lyapunov methods to choose parameters for the state regulator such that

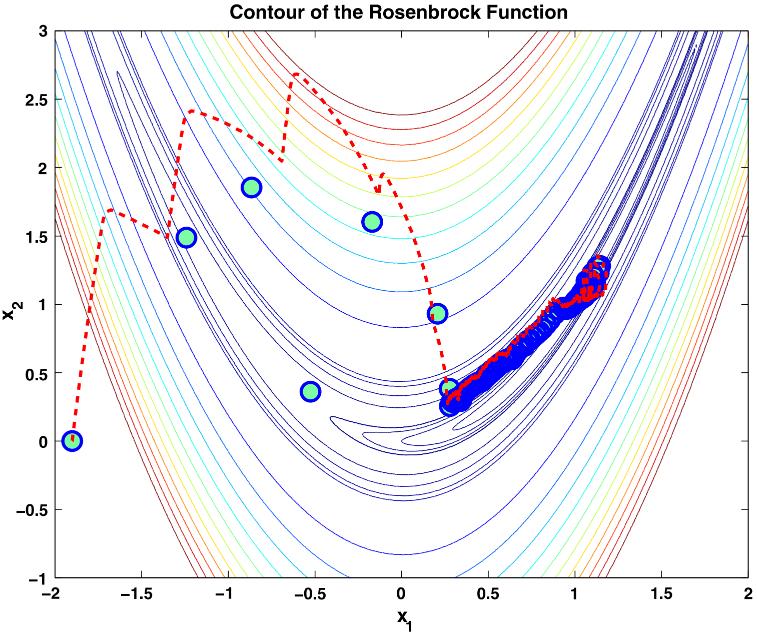


Fig. 5.10 Direct adaptive control for unknown plant dynamics with unknown unbounded input disturbance, $k_1 = 2$, $k_u = 2$, $p_u = 3$, $\alpha_k = 0.004$, $g_0 = 0.9$, $B_1 = 0$, $W_u = 20$, $\gamma_u = 5$: phase portrait

x will not only converge to the required vicinity of x^s_{k+1}, but also not violate the constraints and stay within the require set S_x during the transient.

The NOESC framework allows for large design flexibility for more general nonlinear systems and various performance functions. For example, we can easily extend the result to input-output feedback linearizable systems (similar to Sect. 4.1.3), or other nonlinear systems as long as we can design a robust state regulator. Moreover, since the reference signal $r_k(t)$ is also periodic, it maybe possible to apply iterative learning control [4] to improve the tracking performance. Furthermore, by using a non-derivative optimization algorithm such as the trust region method [8, 9] or gradient estimation, we can remove the assumption of gradient measurement.

Future research will include the design of output feedback state regulator, and will explore other robust numerical optimization algorithms. The recent paper [3] successfully combines the state regulator with the Recursive Smith-Power (RSP) algorithm, which only requires performance output measurements. This is made possible by the fact that RSP algorithm can be reduced to a sequence of one dimensional optimization routines, and the state can be controlled to move freely along a straight line (point mass dynamics) to achieve optimization objectives. It points to a great research direction focused on how to merge the optimization algorithm with the controller design.

References

1. Ariyur, K.B., Krstić, M.: Real-time Optimization by Extremum-seeking Control. Wiley-Interscience, Hoboken (2003)
2. Khalil, H.K.: Nonlinear Systems. Prentice Hall, Upper Saddle River (2001)
3. Mayhew, C.G., Sanfelice, R.G., Teel, A.R.: Robust source-seeking hybrid controllers for autonomous vehicles. In: Proceedings of the American Control Conference, pp. 1185–1190 (2007)
4. Moore, K.L.: Iterative learning control: An expository overview. In: Applied and Computational Controls, Signal Processing, and Circuits, vol. 1(1), pp. 425–488 (1998)
5. Nocedal, J., Wright, S.: Numerical Optimization. Springer, Berlin (1999)
6. Spooner, J.T., Maggiore, M., Ordóñez, R., Passino, K.M.: Stable Adaptive Control and Estimation for Nonlinear Systems, Neural and Fuzzy Approximator Techniques. Wiley, New York (2002)
7. Teel, A.R., Popović, D.: Solving smooth and nonsmooth multivariable extremum seeking problems by the methods of nonlinear programming. In: Proceedings of the American Control Conference, vol. 3, pp. 2394–2399 (2001)
8. Zhang, C., Ordóñez, R.: Non-gradient extremum seeking control of feedback linearizable systems with application to ABS design. In: Proceedings of the Conference Decision and Control, pp. 6666–6671 (2006)
9. Zhang, C., Ordóñez, R.: Robust and adaptive design of numerical optimization-based extremum seeking control. Automatica **45**, 634–646 (2009)

Part II
Applications

Chapter 6
Antilock Braking Systems

6.1 Problem Description

Antilock braking systems (ABS) are an important tool in the automotive industry. They allow the vehicle to stop faster and make safer turns when the wheels are prevented from locking. ABS design was initially proposed to deal with braking on slippery surfaces, i.e., to prevent the wheels from locking and skidding.

Due to the nonlinearity of the dynamics and uncertainty in the braking systems, the design of ABS is difficult. The character of the friction force acting on the tires has a maximum for a low (nonzero) wheel slip and decreases as the slip increases. Standard ABS systems apply braking pressure in a rapid intermittent fashion. In some of them, the purpose of the intermittent action is to "seek" the maximum of the friction characteristic. In this chapter, we study the ABS design via different extremum seeking control schemes; our goal is to design a control algorithm for the braking torque to achieve maximal friction force without prior knowledge of the optimal slip. The wheel model and the perturbation based extremum seeking design are due to Ariyur and Krstić (Chap. 7 of [1]).

Consider the single wheel model depicted in Fig. 6.1. The wheel dynamics are given by

$$m\dot{x}_1 = -N\mu(\lambda), \tag{6.1}$$

$$I\dot{x}_2 = -Bx_2 + NR\mu(\lambda) - u, \tag{6.2}$$

where x_1 is the linear velocity v and x_2 is the angular velocity Ω of the wheel, m is the mass, $N = mg$ is the weight of the wheel, R is the radius of the wheel, I is the moment of inertia of the wheel, Bx_2 is the braking friction torque, u is the braking torque, $\mu(\lambda)$ is the friction force coefficient and the wheel slip λ is defined as

$$\lambda = \frac{x_1 - Rx_2}{x_1} \tag{6.3}$$

for $Rx_2 \leq x_1$.

C. Zhang, R. Ordóñez, *Extremum-Seeking Control and Applications*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2224-1_6, © Springer-Verlag London Limited 2012

Fig. 6.1 The wheel forces

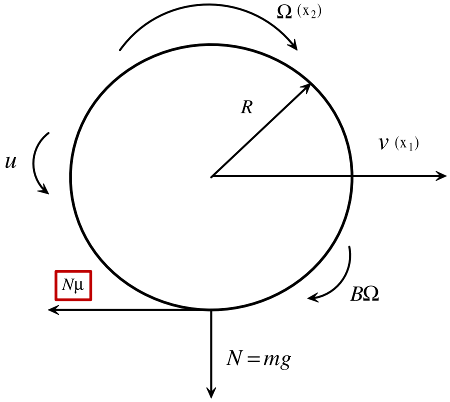
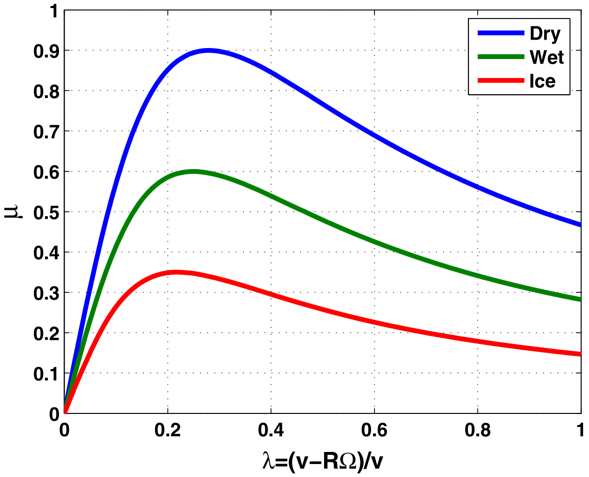


Fig. 6.2 Friction force coefficient



Note that (6.1) and (6.2) are together a fairly accurate model of actual wheel behavior, as they capture the uncertainty due to road and driving conditions inherent in the ABS problem.

There exists a maximum μ^* for the friction force coefficient $\mu(\lambda)$ at λ^*, but λ^* and μ^* will change as the road conditions change. The friction force coefficient $\mu(\lambda)$ is shown in Fig. 6.2 for three road conditions. Now the purpose of the ABS design is to generate a control input u such that the friction force coefficient $\mu(\lambda)$ is maximized, regardless of the road conditions. Moreover, even though the knowledge of $\mu(\lambda)$ is not available, we are able to obtain the measurement of $\mu(\lambda)$ from (6.1), assuming that the linear acceleration $\dot{x}_1$ is measured via an accelerometer.

6.2 Perturbation Based Extremum Seeking Control Design

In order to formulate the problem into the perturbation based extremum seeking setting, let us introduce a constant λ_0 (which is unknown) and define $\tilde{\lambda} = \lambda - \lambda_0$.

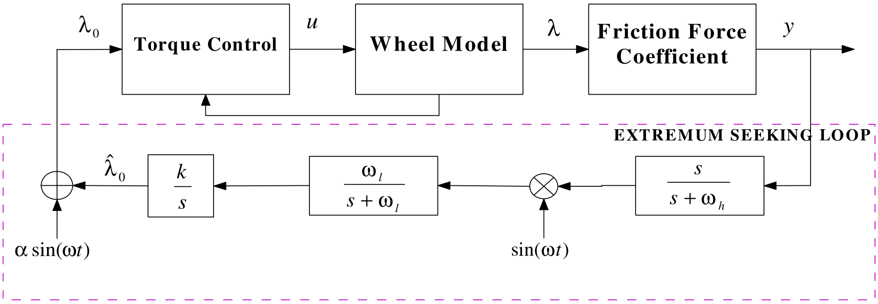


Fig. 6.3 ABS design via perturbation based extremum seeking control

The governing equation for $\tilde{\lambda}$ is

$$\dot{\tilde{\lambda}} = \dot{\lambda} = \left(\frac{Rx_2}{x_1^2} + \frac{mR^2}{Ix_1}\right)\dot{x}_1 + \frac{RB}{Ix_1}x_2 + \frac{R}{Ix_1}u. \tag{6.4}$$

Since $\dot{x}_1$ is measurable via an accelerometer, it is easy to see that the simple feedback linearizing controller

$$u = -\frac{cIx_1}{R}(\lambda - \lambda_0) - Bx_2 - \frac{Ix_2}{x_1}\dot{x}_1 - mR\dot{x}_1, \tag{6.5}$$

where c is a positive constant, makes the equilibrium λ_0 of the system (6.4) exponentially stable, giving

$$\dot{\tilde{\lambda}} = -c\tilde{\lambda}.$$

Note that the control u in (6.5) does not require the knowledge of the unknown function $\mu(\lambda)$ (due to the assumption that $\dot{x}_1$ is available for measurement). Then the wheel model under feedback controller (6.5) can be written as a cascade of input dynamics and a static map:

$$\begin{aligned} \frac{1}{c}\dot{\lambda} &= -(\lambda - \lambda_0), \\ y &= \mu(\lambda). \end{aligned} \tag{6.6}$$

We can apply the perturbation based extremum seeking control scheme given in Fig. 6.3 with

$$\lambda_0 = \hat{\lambda}_0 + \alpha \sin(\omega t).$$

As before, the reader should note that the symbol $\bigoplus$ denotes a summer, whereas the symbol $\bigotimes$ denotes a multiplier.

For simulation purposes, we postulate a simple function that qualitatively matches $\mu(\lambda)$ as in [1]:

$$\mu(\lambda) = 2\mu^* \frac{(\lambda^*)\lambda}{(\lambda^*)^2 + \lambda^2}. \tag{6.7}$$

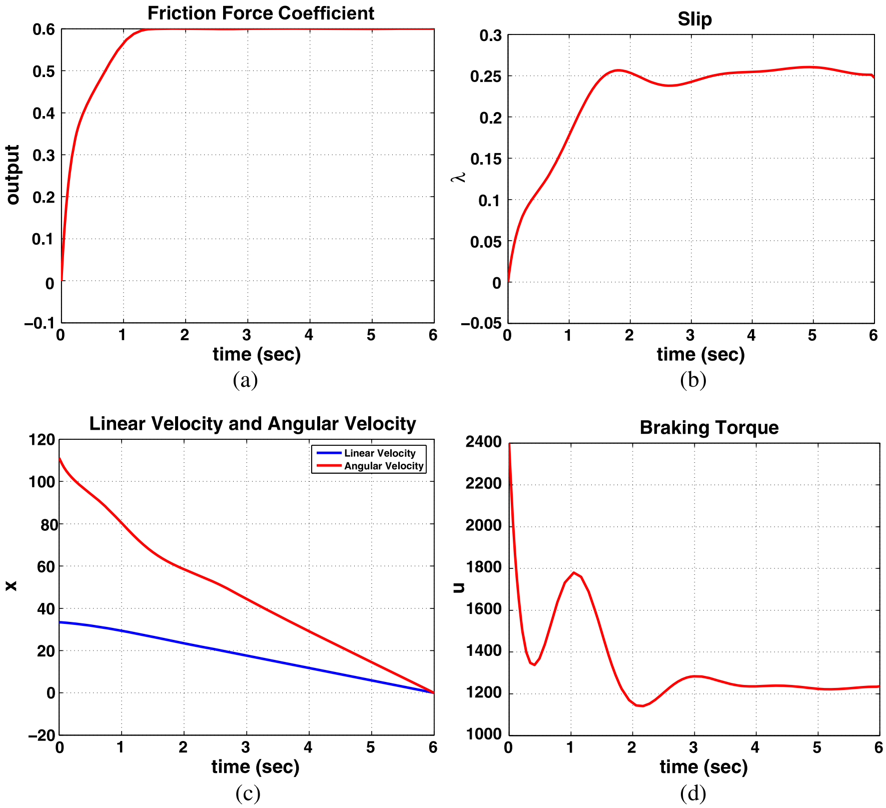


Fig. 6.4 ABS design via perturbation based extremum seeking control: (**a**) friction force coefficient; (**b**) slip; (**c**) linear velocity and angular velocity; (**d**) braking torque

This function has a maximum at $\lambda = \lambda^*$, whose value is $\mu(\lambda^*) = \mu^*$. We run the simulation for $\lambda^* = 0.25$ and $\mu^* = 0.6$. The parameters of the wheel are chosen as $m = 400$ kg, $B = 0.01$ and $R = 0.3$ m. The initial conditions are linear velocity $x_1(0) = 33.33$ m/s, and angular velocity $x_2(0) = 400/3.6$, which makes $\lambda(0) = 0$.

The simulation employs the perturbation based extremum seeking scheme with $\alpha = 0.01$, $\omega = 3$, $\omega_h = 0.6$, $\omega_l = 0.8$, $c = 20$ and $k = 1.5$. For $\lambda_0 = 0.1$, the simulation results are shown in Fig. 6.4. It is seen that during braking, maximum friction force is reached and the car is stopped within the shortest time and distance. The low pass filter in the design can be removed without loss of stability, i.e., $\omega_l = 0$. Its purpose is to attenuate noise in the loop.

6.3 Sliding Mode Based Extremum Seeking Control Design

We can easily use sliding mode extremum seeking to replace the sinusoidal perturbation based extremum seeking scheme. By using the same torque controller (6.5),

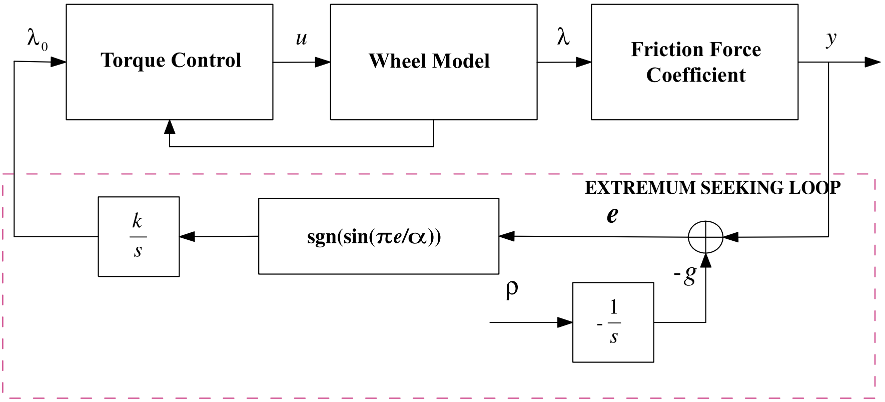


Fig. 6.5 ABS design via sliding mode based extremum seeking control

we apply the sliding mode based extremum seeking method depicted in Fig. 6.5, with

$$\dot{\lambda}_0 = k\, \text{sign} \sin(\pi s/\alpha),$$

where α is a positive constant, s is a switching function defined as

$$e(t) = y - g(t)$$

and $g(t)$ is an increasing function satisfying $\dot{g}(t) = \rho > 0$.

In the simulation, $\mu(\lambda)$ is postulated as in (6.7). The simulation employs the sliding mode based extremum seeking scheme with $\alpha = 0.1$, $\rho = 1$, $c = 20$, and $k = 1.5$. For the same wheel parameters and initial conditions as in Sect. 6.2, the simulation results are shown in Fig. 6.6. It can be seen that during braking, maximum friction force is reached and the car is stopped within the shortest time and distance.

At the same time, note from Fig. 6.6(d) that the braking torque attains large values, and it is highly oscillatory. It is reasonable to speculate that a system such as this may negatively impact ride comfort if implemented in an actual ABS design.

6.4 Numerical Optimization Based Extremum Seeking Control Design

By observing (6.1), we find that x_1 is not controllable from u for a fixed $\mu(\lambda)$. Fortunately, the friction force coefficient μ is only dependent on λ, which is a function of x_1 and x_2, and thus it may be controllable by u. Therefore, we attempt to feedback linearize the system from the input u to output λ.

We define the change of variables

$$\eta = x_1,$$
$$\lambda = (x_1 - Rx_2)/x_1,$$

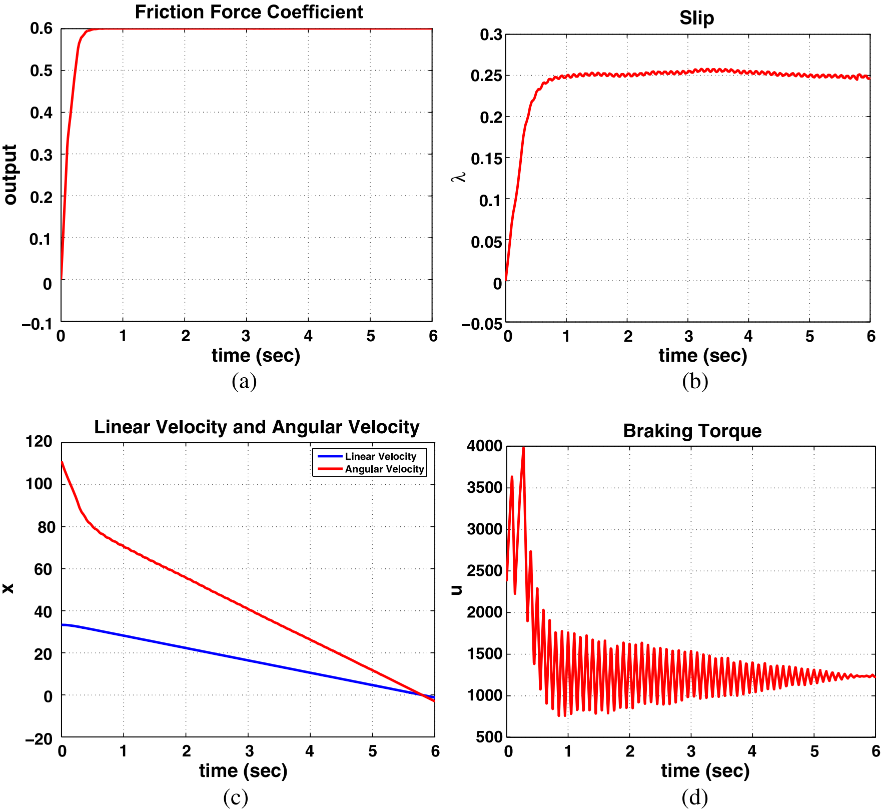


Fig. 6.6 ABS design via sliding mode based extremum seeking control: (**a**) friction force coefficient; (**b**) slip; (**c**) linear velocity and angular velocity; (**d**) braking torque

which transforms the system (6.1), (6.2) into the form

$$\dot{\eta} = -N\mu(\lambda)/m, \tag{6.8}$$

$$\dot{\lambda} = -\left(\frac{R}{Ix_1} + \frac{x_2}{mx_1^2}\right)NR\mu(\lambda) + \frac{R}{Ix_1}(u + Bx_2). \tag{6.9}$$

Since η is the linear velocity, it will be bounded at all times due to physical restrictions. Then, given $x_1(0) > 0$, let the braking torque be

$$u = \frac{Ix_1}{R}(-c\lambda + v) - Bx_2 + \left(\frac{R}{Ix_1} + \frac{x_2}{mx_1^2}\right)IN\mu(\lambda)x_1. \tag{6.10}$$

As we did before, we assume that $\dot{x}_1$ is available for measurement, instead of $\mu(\lambda)$. Thus, using (6.1), the control (6.10) becomes

$$u = \frac{Ix_1}{R}(-c\lambda + v) - Bx_2 + \left(\frac{R}{Ix_1} - \frac{x_2}{mx_1^2}\right)Im\dot{x}_1x_1. \tag{6.11}$$

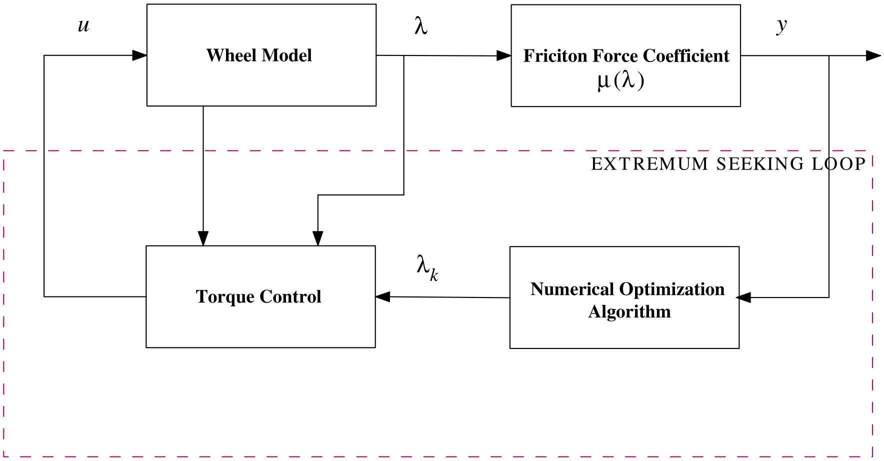
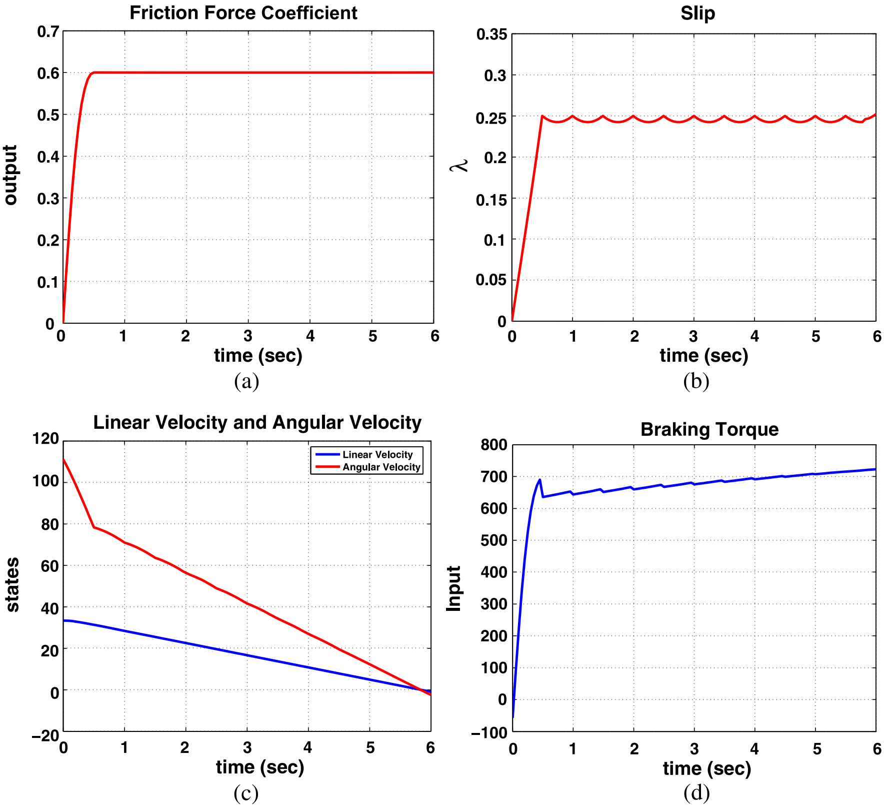


Fig. 6.7 ABS design via numerical optimization based extremum seeking control



(a)

(b)

(c)

(d)

Fig. 6.8 ABS design using line search and finite-time state regulator extremum seeking control: (**a**) friction force coefficient; (**b**) slip; (**c**) linear velocity and angular velocity; (**d**) braking torque

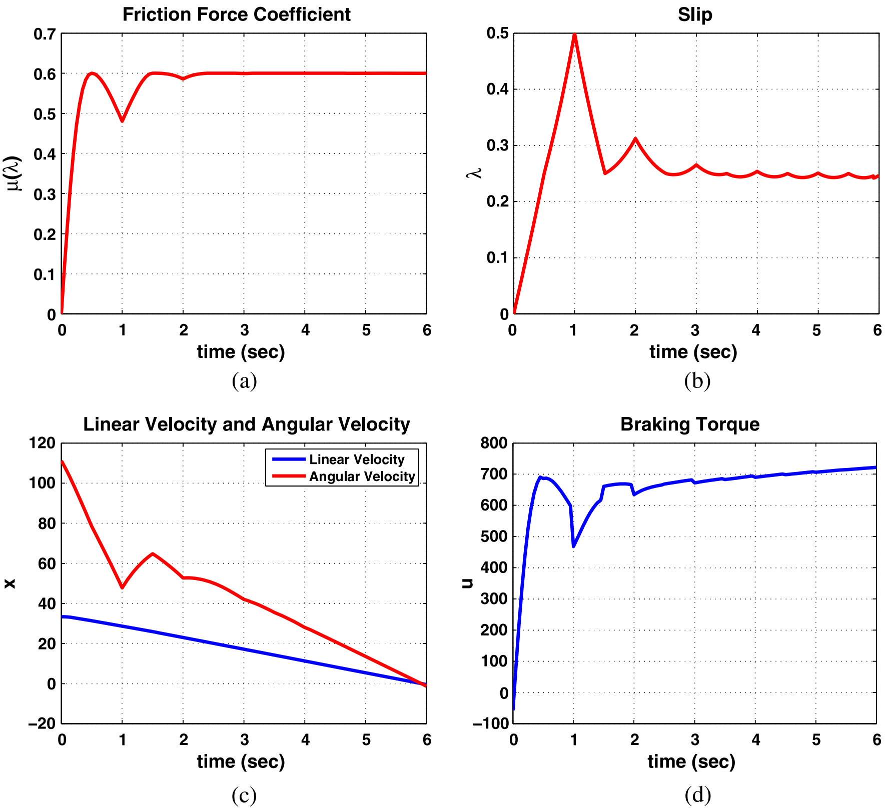


Fig. 6.9 ABS design using trust region and finite-time state regulator extremum seeking control: (**a**) friction force coefficient; (**b**) slip; (**c**) linear velocity and angular velocity; (**d**) braking torque

Then (6.9) becomes

$$\dot{\lambda} = -c\lambda + v, \tag{6.12}$$

where we first investigate the finite-time state regulator of Chap. 4, and thus set v to be the regulator defined in (4.12).

A block diagram of extremum seeking scheme for the wheel model can be found in Fig. 6.7. That is, by designing the control torque as in (6.10), we are able to adjust the slip λ to maximize the friction force coefficient.

In the simulation, $\mu(\lambda)$ is again postulated to be as in (6.7). We use the same wheel parameters and initial conditions as in Sect. 6.2 in the simulations. The simulation results for line search based extremum seeking control are shown in Fig. 6.8, where $c = 1$, and $\delta_k = 0.5$. However, since the finite-time state regulator cannot render the optimal λ^* as an equilibrium point of the closed loop system, we observe that the steady-state slip oscillates.

The simulation results based on trust region method are shown in Fig. 6.9, where no gradient measurements of $\mu(\lambda)$ are needed. As a way to illustrate the robust

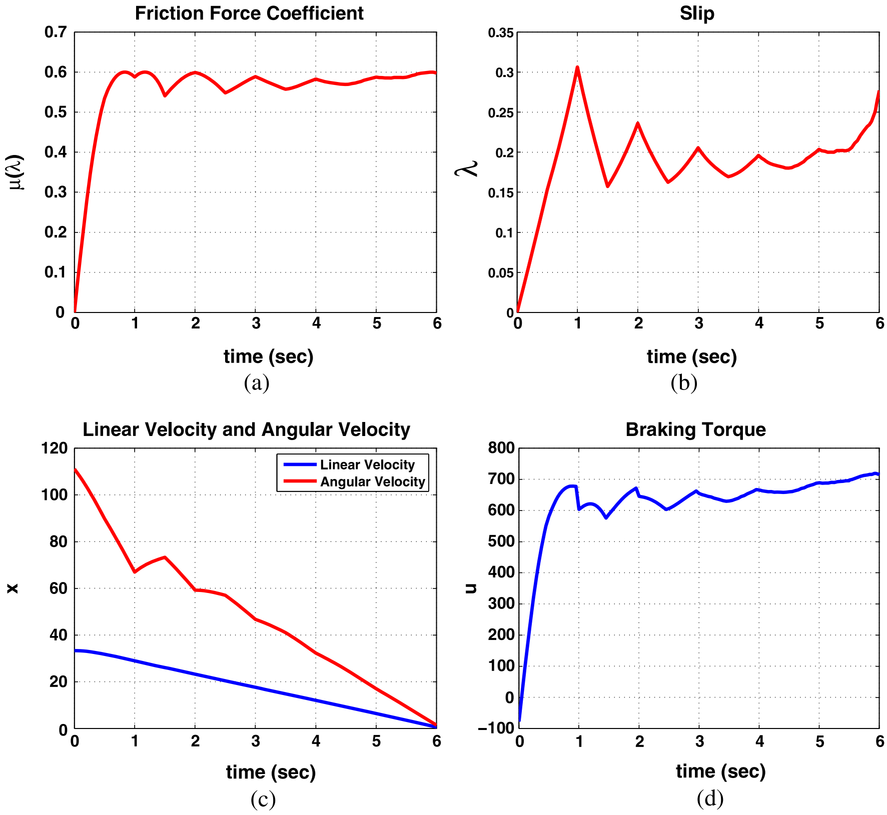


Fig. 6.10 ABS design using trust region and finite-time state regulator extremum seeking control with *input disturbance*: (**a**) friction force coefficient; (**b**) slip; (**c**) linear velocity and angular velocity; (**d**) braking torque

behavior of NOESC using trust region, Fig. 6.10 shows results where the input is disturbed by a uniformly distributed noise with amplitude 2.

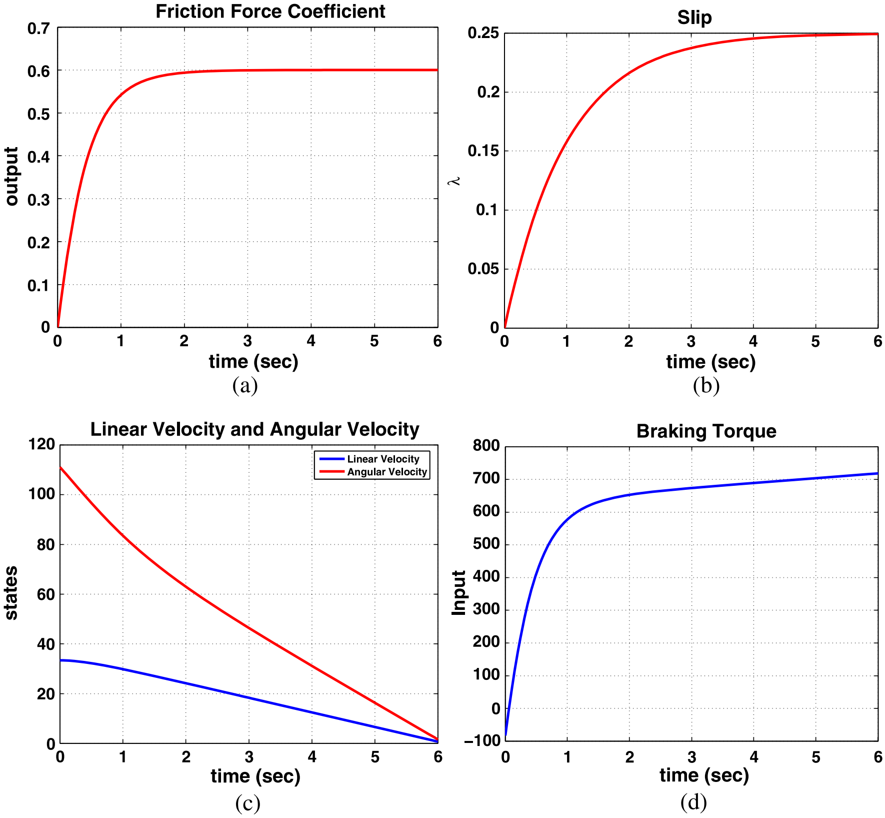
Next, we apply the asymptotic state regulator design from Chap. 5. This time, there is no need to put the design into the finite-time regulator framework since we only have one state to control via one control input. We can render any λ_k to be an equilibrium point for the linearized system in (6.12) by choosing the asymptotic controller

$$v = c\lambda_k,$$

where λ_k is the commanded regulation point from the numerical optimization algorithm, as seen in Fig. 6.7. Let $e = \lambda - \lambda_k$. Then, we have

$$\dot{e} = \dot{\lambda} = c\lambda + v = -c\lambda + c\lambda_k = -ce,$$

which implies e asymptotically converges to zero as long as $c > 0$. It is interesting to note that the control (6.11) from NOESC is identical to the control (6.5) used by



(a)

(b)

(c)

(d)

Fig. 6.11 ABS design using line search and asymptotic state regulator extremum seeking control: (**a**) friction force coefficient; (**b**) slip; (**c**) linear velocity and angular velocity; (**d**) braking torque

PESC, with $\lambda_0 = \lambda_k$. The reason for this is that the PESC ABS controller is in fact a feedback linearizing controller, the same as the NOESC controller.

Now, the simulation results for line search and trust region based extremum seeking design can be found in Figs. 6.11 and 6.12, respectively.

6.5 Conclusions

Since only one parameter needs to be tuned among different extremum seeking control schemes, the perturbation based and sliding mode based designs turn out to be easy to tune in this case. The PESC and SMESC designs do not need to have the gradient; however, they do bring additional oscillations in steady state due to the perturbation signal and the sliding mode, respectively. Further research to deal with oscillations can be found in [3], and time delay can be found in [2].

The design via numerical optimization based extremum seeking can be made gradient free as well using a derivative free trust region method; however, the con-

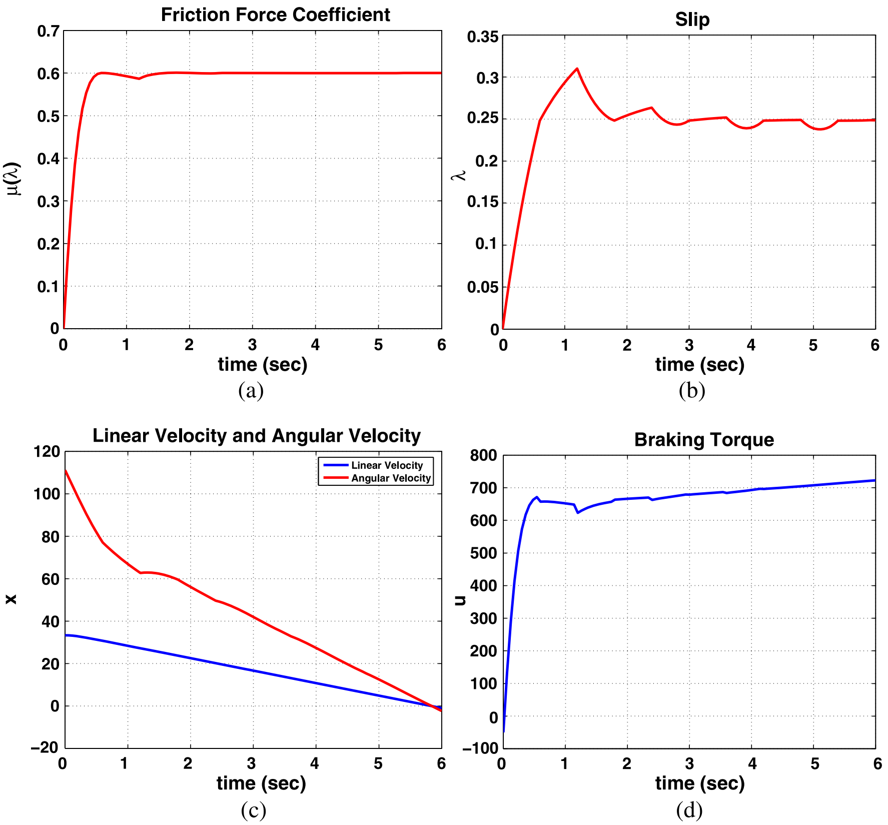


Fig. 6.12 ABS design using trust region and asymptotic state regulator extremum seeking control: (**a**) friction force coefficient; (**b**) slip; (**c**) linear velocity and angular velocity; (**d**) braking torque

vergence may be relatively slow due to the slow convergence of the optimization method. Nevertheless, all NOESC methods achieve brake times comparable with those of PESC and SMESC, with significantly less braking torque required. Oscillation is successfully avoided when using the asymptotic state regulator numerical optimization based extremum seeking control of Chap. 5, because no perturbation signal is used and no sliding mode function is introduced. An additional robustifying term can be easily added to the numerical optimization based design to deal with input disturbances and unmodeled plant dynamics.

References

1. Ariyur, K.B., Krstić, M.: Real-time Optimization by Extremum-seeking Control. Wiley-Interscience, Hoboken (2003)
2. Yu, H., Özgüner, Ü.: Extremum-seeking control strategy for ABS system with time delay. In: Proceedings of the American Control Conference, vol. 5, pp. 3753–3758 (2002)
3. Yu, H., Özgüner, Ü.: Smooth extremum-seeking control via second order sliding mode. In: Proceedings of the American Control Conference, vol. 4, pp. 3248–3253 (2003)

Chapter 7
Impedance Matching in Semiconductor Plasma Processing Chamber

7.1 Introduction to Impedance Matching

Impedance matching is the practice of designing the input impedance of an electrical load or the output impedance of its corresponding signal source in order to maximize the power transfer and minimize reflection from the load. In general, the main reason to do impedance matching is to obtain a more efficient power transfer in a circuit. Moreover, in a complex industrial application, there are cases where the designer needs to interconnect a number of different components into a system, and the only way this interconnection can be performed reliably and predictably is by constraining the reflection coefficients of the various interfaces through impedance matching. Multiple reflections could result in group delay variations that can produce undesired intermodulation in broadband systems [16].

For semiconductor thin film processing applications, impedance matching is used in a semiconductor plasma processing chamber in order to minimize the reflected power back from plasma discharge into the RF cables, and maximize the power transferred from the RF generator into the plasma discharge.

7.1.1 Maximal Power Principle

Whenever a source of power with a fixed output impedance operates into a load (refer to Fig. 7.1), the maximum possible power is delivered to the load when the impedance of the load is equal to the complex conjugate of the impedance of the source. Let

$$Z_l = R_l + jX_l$$

denote the load impedance. The generator (or voltage source) supplies a constant voltage output V and has internal impedance

$$Z_s = R_s + jX_s.$$

C. Zhang, R. Ordóñez, *Extremum-Seeking Control and Applications*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2224-1_7, © Springer-Verlag London Limited 2012

Thus, we have the current

$$I = \frac{V}{Z_s + Z_l}$$

and the voltage

$$V_l = \frac{V}{Z_s + Z_l} Z_l.$$

Then the power delivery to the load is

$$\begin{aligned} P_l &= \frac{1}{4} Re(I V_l^* + I^* V_l) \\ &= \frac{1}{2} |I|^2 R_l \\ &= \frac{1}{2} \frac{|V|^2 R_l}{|Z_s + Z_l|^2}, \end{aligned}$$

where the symbol * denotes complex conjugate. By ensuring that

$$\frac{\partial P_l}{\partial R_l} = 0, \qquad \frac{\partial P_l}{\partial X_l} = 0,$$

the power is maximal transferred to the load if $Z_l = Z_s^*$, that is, by letting

$$\begin{aligned} R_l &= R_s, \\ X_l &= -X_s, \end{aligned}$$

and one obtains the value of the maximum power delivered as

$$P_{l,\max} = \frac{|V|^2}{8 R_s}.$$

Moreover, the quantity

$$\Gamma = \frac{Z_l - Z_s}{Z_l + Z_s}$$

is defined as the *reflection coefficient* and is generally a complex number.

7.1.2 Reflected Power

In the Radio Frequency (RF) domain, if at any point on the transmission line one inserts a directional coupler sampling the voltage and current, it will produce two quantities called the forward and reflected voltage. These voltages are the amplitudes of the two traveling waves and together make up the standing wave on the

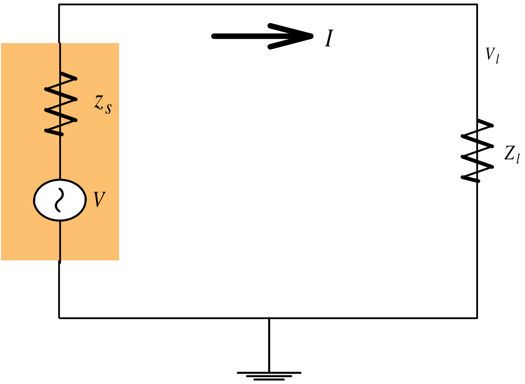


Fig. 7.1 A simple source load circuit

line. One can define the *forward voltage* as

$$V_f = (V + Z_0 I)/2$$

and the *reflected voltage* as

$$V_r = (V - Z_0 I)/2,$$

where the quantity Z_0 is arbitrary and it could be the characteristic impedance of the transmission line [11]. Then, the *forward power* is defined as

$$P_f = |V_f|^2/Z_0$$

and the *reflected power* is given by

$$P_r = |V_r|^2/Z_0.$$

The reflected power could be as low as zero when $V_r = 0$ as the impedance is matched. The difference between the two powers is called *load power*, that is,

$$\begin{aligned} P_l &= P_f - P_r \\ &= \frac{1}{Z_0}\left(|V_f|^2 - |V_r|^2\right) \\ &= \frac{|V_f|^2}{Z_0}\left(1 - |\Gamma|^2\right), \end{aligned} \tag{7.1}$$

where we further express the reflection coefficient as $\Gamma = \frac{V_r}{V_f}$.

An alternative metric is the *voltage standing wave ratio* (*VSWR*), defined as

$$\Psi = \frac{1 + |\Gamma|}{1 - |\Gamma|}.$$

Consider for example a 50 Ω source impedance. Then, the VSWR $\Psi = 1$ if $Z_0 = 50$, and $\Psi = \infty$ if $Z_0 = 0$ or $Z_0 = \infty$ (that is, either a closed or open circuit). Thus, the main goal for RF impedance matching is to maximize the forward power to the load or minimize the reflected power, which is equivalent to reducing the reflection coefficient Γ to 0 or VSWR Ψ to 1.

7.1.3 Impedance Matching Techniques and Challenges in the Semiconductor Plasma Processing Chamber

Plasma processing technology is widely used in the semiconductor wafer processing. Plasma processing is also critical in the aerospace, automotive and waste management industries [5]. In the semiconductor industry, RF driven plasma discharge is commonly used in Etch or Chemical Vapor Deposition (CVD) processes.

An idealized semiconductor plasma processing chamber/equipment contains two planar electrodes separated by a certain gap and driven by an RF power source. The processing substrate is placed on one electrode, chemical gases flow through to form discharge, and effluent gases are removed by a vacuum pump. One of most important design requirements for the plasma chamber is to maximize the power transferred from the RF generator to the plasma discharge load, or to minimize the amount of power reflected back from the plasma discharge into the RF generator.

The schematic of a typical RF power delivery system in the plasma processing chamber can be found in Fig. 7.2, where a generator is connected through RF cables to an RF matching network. This network is generally part of the plasma processing chamber. When the impedance is mismatched, the high VSWR could affect the power accuracy of the generator and thus impact the process repeatability.

The current generation of generators can operate at “LOAD” mode, that is, closed-loop control on the delivery power to ensure the process repeatability. In other words, the plasma discharge always obtains the same power, no matter how large the impedance mismatch is. Mismatched loads that are run for extended periods of time can damage the RF delivery system, for instance due to overheating of the cables and connectors because of an increase in the RF current. Mismatched loads can also cause arcing in the component due to high voltage. Moreover, the amount of electromagnetic interference increases under mismatched conditions. There is also a significant cost benefit in using the proper size of generator if the reflected power is minimized, because one does not have to increase the power limit of the generator in order to compensate for additional reflected power.

There are generally two main impedance matching techniques utilized in the semiconductor industry. One is called *automatic matching network*, where physical components like “loss-less” passive electrical elements (such as capacitors and inductors) are used. This physical matching network is placed in between the generator and the plasma processing chamber, and it is placed as close as possible to the chamber in order to minimize further transmission line loss. A practical matching network is shown in Fig. 7.3, where impedance matching is achieved by varying the values of the capacitors or inductors within the L type of matching network, such

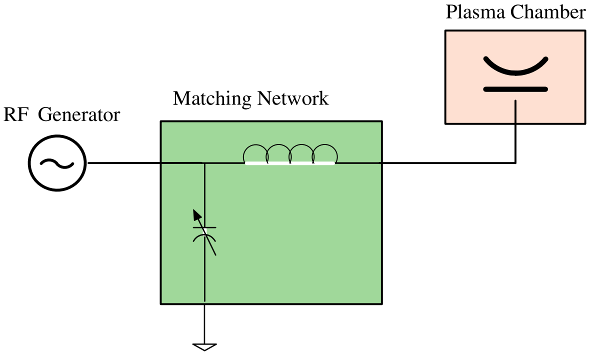


Fig. 7.2 An RF power delivery system of a plasma processing chamber

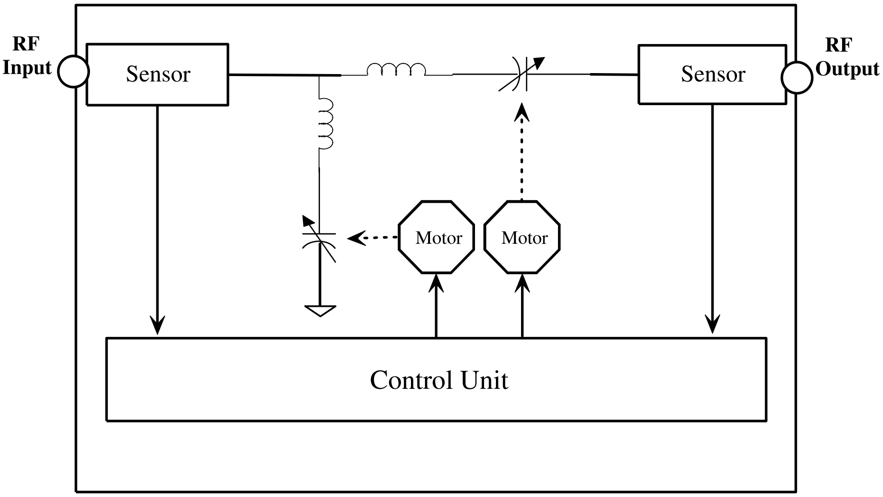


Fig. 7.3 A practical matching network schematic diagram

that the load impedance from the port of the generator equals the generator internal impedance. Variable capacitors are commonly used instead of variable inductor, as variable capacitors are of much more compact design and lower cost.

The second technique is the so-called *generator frequency tuning*, where the RF frequency at the generator end is varied in order to reduce the reflected power back to the generator. That is, the impedance is changed by tuning the frequency ω of the generator. A directional coupler [6, 7] is generally available in the RF generator, which provides measurements of the reflected power and is used as feedback to control the frequency. There are cases when both techniques are used together in order to obtain a larger window of tune space; that is, in order to satisfy a broad range of plasma loads [3].

Moreover, there are two modes to tune or utilize the matching network in order to achieve maximal power delivery: the "auto-tune" mode and the "preset" mode. These modes correspond to closed-loop and open-loop control, respectively. In the "auto-tune" mode, the variable capacitor is adjusted automatically during the plasma processing in order to achieve the lowest reflected power. In the "preset" mode, the variable capacitor value is pre-determined and is fixed during the plasma processing.

There are advantages and disadvantages to these two methods, keeping in mind that from the industrial manufacturing point of view, process repeatability and reproducibility are the number one priority, even compared with achieving the lowest reflected power. For the "auto-tune" mode, it is generally implemented by feedback control of the variable capacitors, based on reflected power reading or phase/magnitude sensor inputs. The advantage is that it can compensate for the drifting of load impedance over the production life time. However, in certain cases, such as when plasma arcing occurs or during process recipe[1] transition from one step to another, a dramatic change of plasma impedance is expected and improper tuning could further aggravate arcing. The "preset" mode, on the other hand, does yield better repeatability. The challenge with this mode is to devise the desired preset in the first place, since this task requires the knowledge of load impedance (plasma impedance) and/or a significant amount of trial and error.

A similar idea is found in [8, 13], where deterministic tuning is described if one knows the load impedance, and functional tuning is basically iterative optimization via the directional coupler sensor measurements. Moreover, a genetic algorithm is also studied for impedance matching in [14]. A reflection coefficient with gradient type search is combined to achieve impedance matching in [9]. For matching networks having phase and magnitude sensors, commonly the phase sensor is used together with a PID controller to adjust the tune capacitor, and the magnitude sensor is used to control the load capacitor. Also, voltage and current sensors are used in the matching network in order to provide sensor measurements to control the shunt and series capacitors [15]. This approach is widely used today by many companies providing automatic matching network products, such as Seren Industrial Power Systems [12], Daihen Advanced Component [4], Advanced Energy Industries [1], etc.

If the exact load impedance is known, the "auto-tune" and "preset" modes are the same, since the system can be operated in open-loop. The main challenge is the unknown load impedance case. The load impedance is mainly composed of plasma impedance and other impedances due to hardware connections, cables, connectors and installations. Firstly, the load impedances are difficult to model, because plasma discharge impedance is a function of the particular process recipe running on the chamber (for example, consider the experimental results described at the end of the chapter and shown in Fig. 7.15(d), where the reflected power changes as pressure oscillates). Furthermore, plasma impedance may even change during the processing of the substrate wafer itself, even if all the other process conditions remain the same. In Fig. 7.4, one can observe the DC bias voltage changing during the wafer processing, which specifically implies that the chamber impedance is changing. This can be explained by considering that the DC bias voltage is mainly impacted by the

[1] A process recipe is a predefined chamber operation conditions, such as gas mixture amount, chamber pressure, temperature and RF power, etc.

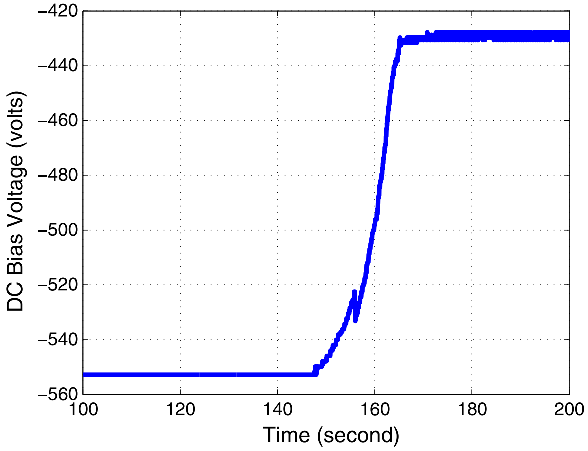


Fig. 7.4 The DC bias voltage changes during the Etch process

chamber impedance, and this impedance change mainly arises from the wafer film composition changes during the Etch process. Moreover, other load impedances from hardware may also slightly differ from chamber to chamber and are sensitive to chamber condition, mechanical tolerances and installation.

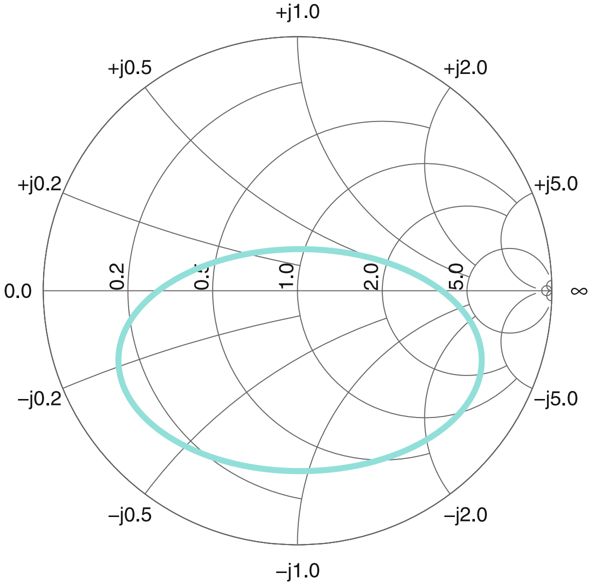
Other matching network design challenges include the range of the variable capacitor needed to cover all possible load impedances, commonly referred to as the matching network tune space. This tune space is generally simulated first, and then a wide selection of process recipes are tested to finalize. The Smith chart is one of the most widely used engineering tools in the earlier stage of matching network design and is still a powerful tool today for analysis. A typical tune space in Smith chart can be found in Fig. 7.5, where the light blue circle area is one illustrative example of the tune space. In the context of discussion of tuning algorithms, we always assume the load impedance is well within the tune space, i.e., the system is controllable.

In this chapter, we will present two tuning algorithms applied to RF matching network design for a capacitive coupled plasma (CCP) chamber: the first algorithm's objective is to improve the productivity by providing a tuning method in order to give the user an optimized preset for each particular recipe; the second one is a real time auto-tuning algorithm.

7.1.4 Plasma Load Impedance Estimation for a CCP Chamber

A CCP chamber can be modeled as two capacitors (the sheath capacitors C_{s_1}, C_{s_2}) serially connected to an RLC circuit (L_0 and R_0 are serially connected, and then parallel connected with C_0). The circuit model can be found in Fig. 7.6, where an RF generator and an LC-type matching network are connected with a CCP chamber. Again, the goal is to tune the matching network inductor and capacitor such that

Fig. 7.5 Tune space in Smith chart representation



$Z_S = Z_P^*$, therefore maximizing the delivery power. Here, we make the simplifying assumption that the load impedance is the same as the plasma impedance.

Let f be the RF frequency in Hz and $\omega = 2\pi f$ (rad/s). Thus, we have

$$C_0 = \frac{\varepsilon_0 A}{d}, \tag{7.2}$$

$$L_0 = \frac{1}{\omega_{pe}^2 C_0}, \tag{7.3}$$

$$R_0 = \nu_m L_0, \tag{7.4}$$

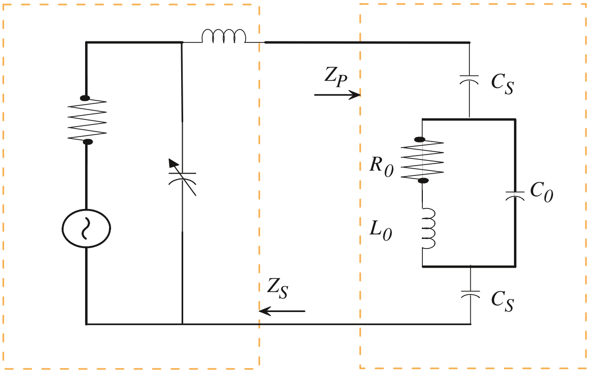
where

- $\varepsilon_0 = 8.8542 \times 10^{-12}$ (F/m) is the permittivity of free space,
- A is the electrode area,
- d is the gap between the two electrode and generally $d \ll A/2\pi$,
- $\omega_{pe} = \sqrt{\frac{e^2 n_0}{\varepsilon_0 m}}$ is the electron plasma frequency (rad/s),
- e is the electron elementary charge (1.6022×10^{-19} C),
- m is the electron mass (9.1095×10^{-31} kg),
- n_0 is the plasma density (m^{-3}), and
- ν_m is the electron-neutral collision frequency for momentum transfer.

Moreover, we assume the two sheath capacitances are equal, or $C_{s_1} = C_{s_2} = C_s$, and

$$C_s = 1.226 \frac{\varepsilon_0}{s_m}, \tag{7.5}$$

Fig. 7.6 A circuit model for a CCP chamber



where s_m is the sheath thickness. At steady state, if we assume a collision frequency ν_m and an electron temperature T_e, we are able to calculate the plasma density n_0 given the chamber pressure p and the RF forward power P_f; then we can calculate the sheath thickness and sheath capacitance, and finally we can estimate the plasma impedance as

$$\begin{aligned} Z_p &= R_p + jX_p \\ &= \frac{2}{j\omega C_s} + \frac{(j\omega L_0 + R_0)/(j\omega C_0)}{j\omega L_0 + R_0 + 1/(j\omega C_0)}. \end{aligned} \tag{7.6}$$

In some simple cases, we may be to obtain the collision frequency ν_m and electron temperature T_e given the knowledge of pressure and RF power, and we can estimate the above chamber impedance Z_p in an easy fashion or using (7.6). Consider for example a simple argon discharge system. In this case we have the neutral gas density

$$n_g = \frac{p}{kT_g}, \tag{7.7}$$

where $k = 1.3807 \times 10^{-23}$ J/K, p is pressure in Pa, and T_g is the gas temperature in kelvin (K). We know

$$K_{iz} = \frac{2}{n_g d} u_B \; (\mathrm{m}^3/\mathrm{s}),$$

where K_{iz} is the reaction rate for ionization, and

$$u_B = \sqrt{\frac{eT_e}{M}}$$

is the Bohm velocity and M is the ion mass (the atomic mass unit of Ar is 39.95, therefore $M = 39.95 \times 1.67 \times 10^{-27}$ kg). Then, we can compute

$$K_{iz} = 2.34 \times 10^{-14} T_e^{0.59} \exp(-17.44/T_e),$$

so we can solve the following equation for T_e,

$$2.34 \times \left(e^{-14}\right) T_e^{0.59} \exp(-17.44/T_e) = \frac{2}{n_g d}\sqrt{\frac{eT_e}{M}}. \tag{7.8}$$

Then, for Ar discharge, we know

$$K_{el} = 2.336 \times 10^{-14} {T_e}^{1.609} \exp\left(0.0618(\log T_e)^{-}0.1171(\log T_e)^3\right),$$

where K_{el} is the reaction rate for elastic scattering, and then we can obtain

$$\nu_m = K_{el} n_g. \tag{7.9}$$

Let the total energy loss be given by

$$E_t = (K_{iz}\xi_{iz} + K_{ex}\xi_{ex} + K_{el} 3mT_e/M)/K_{iz} + 7.2T_e, \tag{7.10}$$

where

- $\xi_{iz} = 15.76$ eV is the ionization energy,
- $\xi_{ex} = 12.14$ eV is the excitation energy, and
- $K_{ex} = 2.48 \times (10^{-14}) T_e^{0.59} \exp(-17.44/T_e)$ is the reaction rate for excitation.

The first term in (7.10) is the effective energy loss per electron-ion pair, whereas the $7.2T_e$ term is the electron energy loss from the plasma.

Now, from energy balance the plasma density can be computed as

$$n_0 = \frac{1}{2}\left[\frac{m\left(\nu_m d + 2\sqrt{\frac{8eT_e}{\pi m}}\right)}{e^3 u_B (2T_e + 7.2T_e)}\right]^{1/2} J_1. \tag{7.11}$$

Then, given forward power P_f, one can solve the following equation for the current flux J_1,

$$P_f = 2eAn_0 u_B E_t + 1.5 u_B J_1^2/\left(\varepsilon_0 \omega^2\right), \tag{7.12}$$

where $\varepsilon_0 = 8.8542 \times 10^{-12}$ (F/m) is the electric constant. Once we have solved for current flux J_1 from (7.12), the plasma density can be calculated from (7.11), and the sheath thickness can be obtained with

$$s_m = \frac{J_1}{e\omega n_0}, \tag{7.13}$$

and finally the plasma impedance is computed as

$$\begin{aligned} Z_p &= R_p + jX_p \\ &= \frac{2P_f}{(J_1 A)^2} - j\frac{4s_m}{\omega\varepsilon_0 A}. \end{aligned} \tag{7.14}$$

Now, given an L type RF matching network as seen in Fig. 7.6, one can easily calculate the desired inductance as

$$L_M = \left(\sqrt{R_p R_s - R_p^2} - X_p\right)/\omega, \tag{7.15}$$

where R_s is the generator resistance, and normally $R_s = 50\ \Omega$, and the desired capacitance is obtained with

$$C_M = \sqrt{\frac{1}{R_s R_p} - \frac{1}{R_s^2}}/\omega. \tag{7.16}$$

Example 7.1 Consider a capacitively coupled plasma system operating at 5 mTorr pressure with argon only and 1000 W RF power. The discharge gap is 5 cm, the electrode areas are 1000 cm^2, and the RF frequency is 13.56 MHz. Then, based on the above analysis, one can easily calculate the plasma impedance

$$R_D = 1.61, \qquad X_D = -123.56.$$

Therefore, the desired capacitance is approximately 1284 pF, and the inductance is approximately 1.554 µH.

7.2 Impedance Matching via Extremum Seeking Control

Now, we can express the plasma load impedance as a function of chamber pressure p, RF frequency ω, electron temperature T_e, collision frequency ν_m, plasma density n_0, etc., as

$$Z_p = f_p(T_e, \nu_m, n_0, \theta),$$

where f_p is some unknown nonlinear function, and θ is a parameter array that could contain all known values in the recipe, such as $\theta = [p, \omega, T_g, d, M]^\top$.

Ideally, if one could feed back the load impedance, then the impedance matching becomes an easy set-point regulation problem. In reality, it is impossible to feed back the load impedance in real time. As seen from the analysis in Sect. 7.1.4, even for a simple Ar discharge system, it is fairly complicated to estimate the plasma load impedance. Firstly, it is only done at steady state and secondly, we have to assume some knowledge about cross section and other important parameters. For actual process recipes, the chemistry is much more complicated than Ar (it will typically be a mix of four or five gases), and there are multiple frequencies ongoing at the same time. Also, the load impedance is not just the plasma impedance itself, but the actual hardware design and chamber condition change over time will all contribute as part of the load impedance. The discussion presented thus far is only a starting point to provide an approximate understanding of what the plasma impedance actually is for different process conditions, and it allows one to design the matching network to have proper tune space to cover all the possible plasma impedances. Then, the non-model based extremum seeking control plays an important role, because it can bypass the dependency on exact knowledge of plasma impedance and it can directly use the knowledge of reflected power to achieve the impedance matching. Of course, an approximate estimation of the load impedance typically can provide good initial conditions for the tunable capacitors, thus potentially helping improve the transient performance.

The reflected power is a function of the load impedance Z_p and the matching network tunable capacitors impedance Z_{c1}, Z_{c2} (typically, one or two variable capacitors are used),

$$P_r = f_r(Z_p, Z_{c1}, Z_{c2}),$$

where P_r is the reflected power and f_r is an unknown nonlinear function. In general, P_r is measured at the RF generator via the directional coupler. Here, we use reflected power reading from the generator to control the tunable capacitor, instead of adding another phase/magnitude sensor in the matching network, which is the common industrial practice. This significantly reduces the cost, simplifies the matching network design and improves the quality and reliability. Therefore, the goal is to design a control law to tune the variable capacitors such that the reflected power is minimized for an arbitrary load impedance, assuming this load impedance is within the matching network tune space.

This problem statement exactly falls into the framework of extremum seeking control. Moreover, as the variable capacitor is motor-driven and generally integrated as a subcomponent in the matching network, this means that the system itself is asymptotically stable, or the state regulator is already achieved by the motorized capacitors. All we need to focus on is how to generate the desired capacitance trajectory for the variable capacitors to follow.

7.3 Dual Frequency Matching Network Tuning via Direct Search Based NOESC

In this application, a particular type of NOESC approach is taken to solve the impedance matching network tuning problem. The matching network is already designed to achieve a fixed preset functionality for a low and high frequency RF bias delivery [18, 20]. The matching network is well designed, such that the crosstalk between the two frequencies is minimized. Thus, we can treat the reflected power minimization of two frequencies as two one-dimensional extremum seeking control problems. The schematic diagram of this system can be found in Fig. 7.7.

First, we need to investigate controllability of the system. In order to do so, a manual scan of reflected power versus capacitor value is carried out to investigate the tuning capability. A particular process condition is chosen with 4500 W low frequency power and 1000 W high frequency power. The scan of low frequency matching network can be found in Fig. 7.8, and the scan for high frequency matching

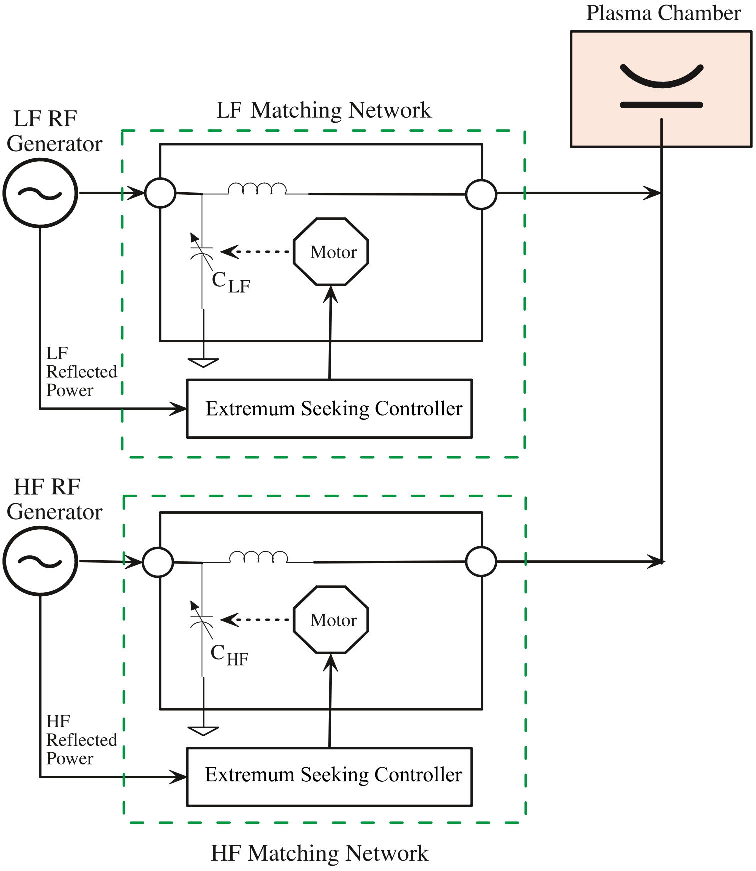


Fig. 7.7 The dual frequency matching network

network can be found in Fig. 7.10. The variable capacitance is represented here as percentage and the scan is done with 5% increment steps.

One can clearly see in Fig. 7.9 that for low frequency RF delivery, the lowest reflected power is achieved for the capacitor C_{LF} between 50% and 55%. For high frequency bias, the optimized preset value for C_{HF} is between 30% and 35%, as can be seen in Fig. 7.11. Thus, one can conclude that for this particular process, the matching networks are capable of tuning down the reflected power. That is, the load impedance is well within the matching network tune space. Also, both low and high frequency generators have frequency tuning capability, and the general settling time of frequency tuning is about 1 second (refer to Figs. 7.9 and 7.11).

As the matching network already have well designed motor-driven vacuum capacitors, in the context of NOESC framework, the state regulator is already available. Thus, we just need to focus on the design of the optimization algorithm. As

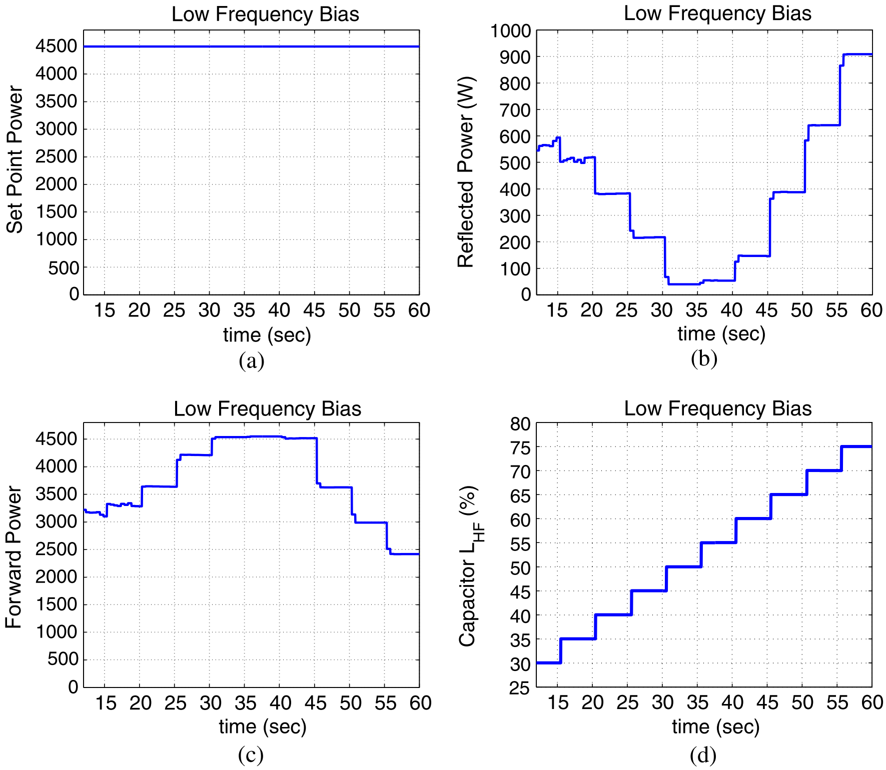


Fig. 7.8 Manual scan low frequency RF delivery: (**a**) set point; (**b**) reflected power; (**c**) forward power; (**d**) low frequency tune capacitor value

the derivative signal is difficult to obtain in a this case, a direct search[2] approach is taken here instead of a more complex numerical optimization method:

1. Start with a preset capacitor set point $C(0)$ based on existing best known value or user input.
2. Increase or decrease the capacitor set point by a step $C(k+1) = C(k) \pm \Delta C$, wait for certain time T_w, read the reflected power P_f.
3. If none of the two capacitor set-point steps yields a reduction in the reflected power, set $\Delta C = \Delta C/2$ and go back to Step 2.

Such a simple direct search algorithm combined with the well designed matching network motorized capacitors achieve great performance in practice. In order to simplify the control design and avoid crosstalk between the frequency tuning

[2] Direct search is used here as a valid optimization method for NOESC because its behavior and properties are similar to those of derivative-free trust region methods. Moreover, since we rely on an already existing motor controller, all stability, convergence and robustness results from Chaps. 4 and 5 directly apply here.

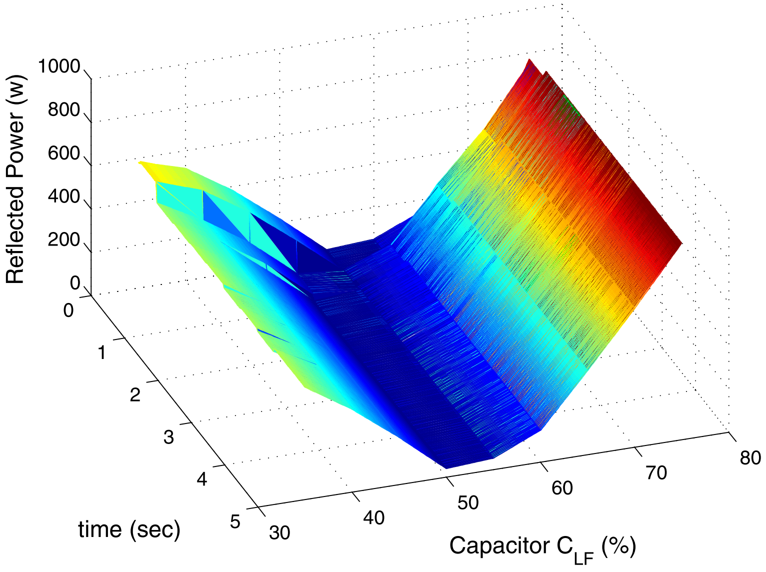


Fig. 7.9 Low frequency RF bias reflected power versus matching network tune capacitor C_{LF}

and capacitor tuning, we have included the waiting time T_w to be 2 seconds. Also, the high frequency generator is configured in LOAD mode; that is, the generator is in closed-loop control on the load power (the difference between forward and reflected power). In this manner, the generator will ensure that the load power equals the set point. Therefore, as one can observe in Fig. 7.10, the forward power minus the reflected power equals the set point, while this is not the case for low frequency generator, as shown in Fig. 7.8. In general, the LOAD mode is preferred in order to ensure consistent load power to the plasma chamber for better chamber matching.

Now, we implemented the direct search algorithm to control the matching network variable capacitor position for both low and high frequencies. In particular, the same process is used, and the initial conditions are $C_{LF}(0) = C_{HF}(0) = 20\%$. The actual experimental results can be found in Figs. 7.12 and 7.13, where the optimal capacitance for low frequency is found to be $C_{LF} = 55\%$, and the high frequency capacitance is $C_{HF} = 35\%$. These optimized values confirm the actual capacitor scans performed previously.

The direct search based extremum seeking control software feature has been implemented on more than 200 equipment installations worldwide at different fabrication facilities, where different process applications have been used. This software feature significantly reduces the process development time. And as one can see, the chamber impedance does change over the process time (refer to Figs. 7.12(b) and (d), where the same matching capacitor value ends up with slightly different reflected power) and also drifts over the production time. Therefore, if one inte-

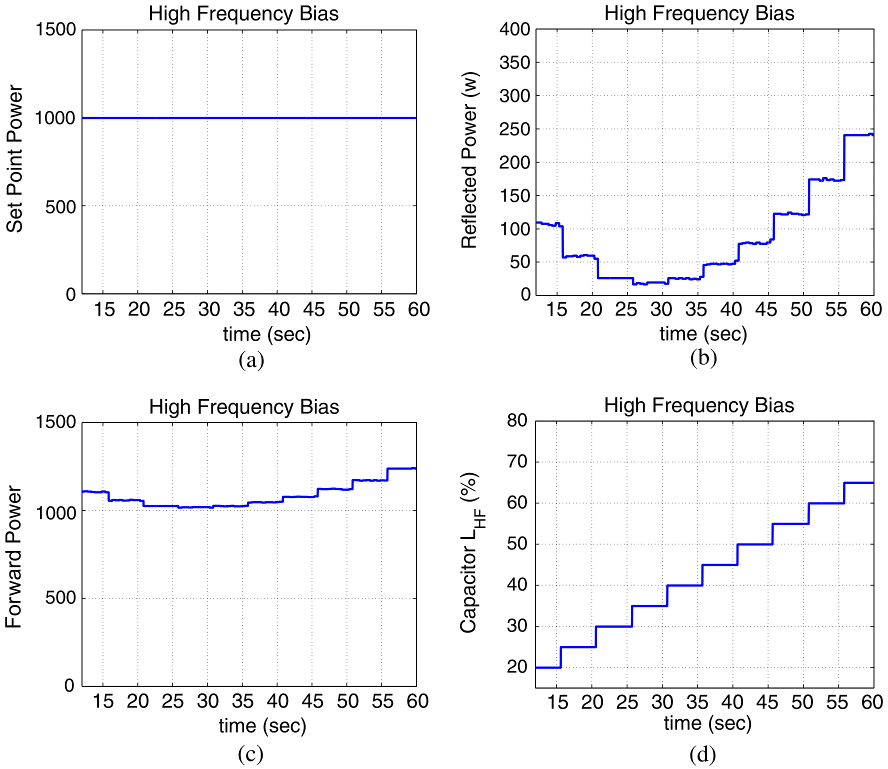


Fig. 7.10 Manual scan high frequency RF delivery: (**a**) set point; (**b**) reflected power; (**c**) forward power; (**d**) low frequency tune capacitor value

grated this feature into the Fab APC (Advanced Process Control) loop, it can further improve run-to-run control performance.

There are also several other practical issues that are covered in the implementation: the reflected power is considered to be minimized if it is less than 5% of the set point; one still prefers to choose a good start tune capacitor value such that the process does not begin with very high reflected power that may lead to high voltage arcing; finally, as the same ESC algorithm is applied to both matching networks, the matching network tuning for each frequency is performed sequentially.

7.4 Dual Capacitor Matching Network Tuning via Perturbation Based Extremum Seeking Control

In this application, a very high frequency (VHF) RF source is connected to the top electrode of the processing chamber. Even without a matching network, the VHF generator has a circulator inside, such that the reflected power can be absorbed

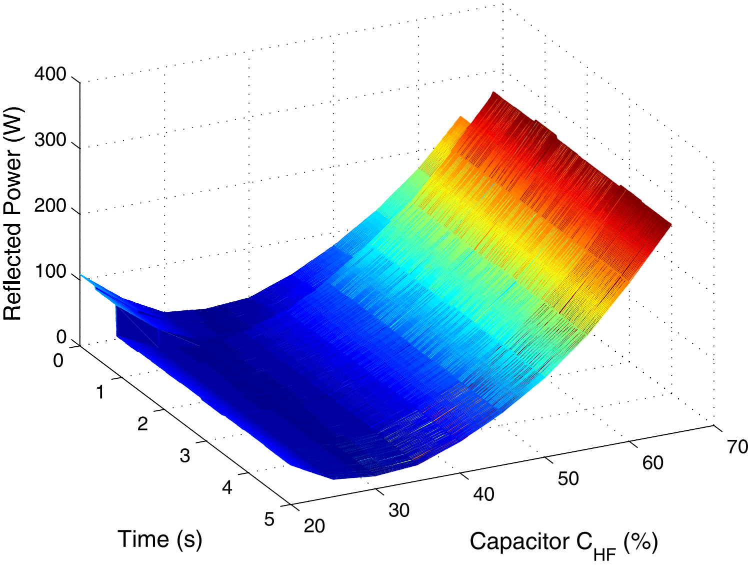


Fig. 7.11 High frequency RF bias reflective power versus matching network tune capacitor C_{HF}

by the generator without damage to the hardware, up to certain level. As processing evolves, a matching network is indeed necessary for some applications [19]. A prototype matching network is developed in-house, with two motorized capacitors (one shunt and one series) used to achieve the desired tune space. Therefore, the extremum seeking control problem becomes a two-input (shunt and series capacitors), one-output (reflected power) problem. Sinusoidal perturbation based extremum seeking control is implemented for this problem. Refer to Fig. 7.14 for a schematic diagram of the system.

A particular process with 2500 W power set point is tested with and without the matching network. The actual experimental results can be found in Fig. 7.15. One can first observe that the reflected power oscillates mainly due to the pressure oscillation. The red curve in Figs. 7.15(d) and (e) (corresponding to the test without matching network) clearly confirms that once the pressure reading is steady the reflected power becomes flat as well, as the pressure significantly impacts the plasma density and therefore the plasma impedance. When the chamber has the matching network installed, the reflected power is significantly reduced, as seen in Fig. 7.15(d). Moreover, as the load impedance is indeed different for the two pressure regions (before and after 80 seconds), the resulting shunt and tune capacitor values are different as well.

Several practical design issues also arise in this problem: the ω_1 and ω_2 sinusoidal perturbation frequencies have to be different, as it is a two-variable SPESC [2, 10] scheme; the time scales for the servo system and the extremum seeking loop need to be separated; finally, the initial conditions and filter parameters are important for the transient performance.

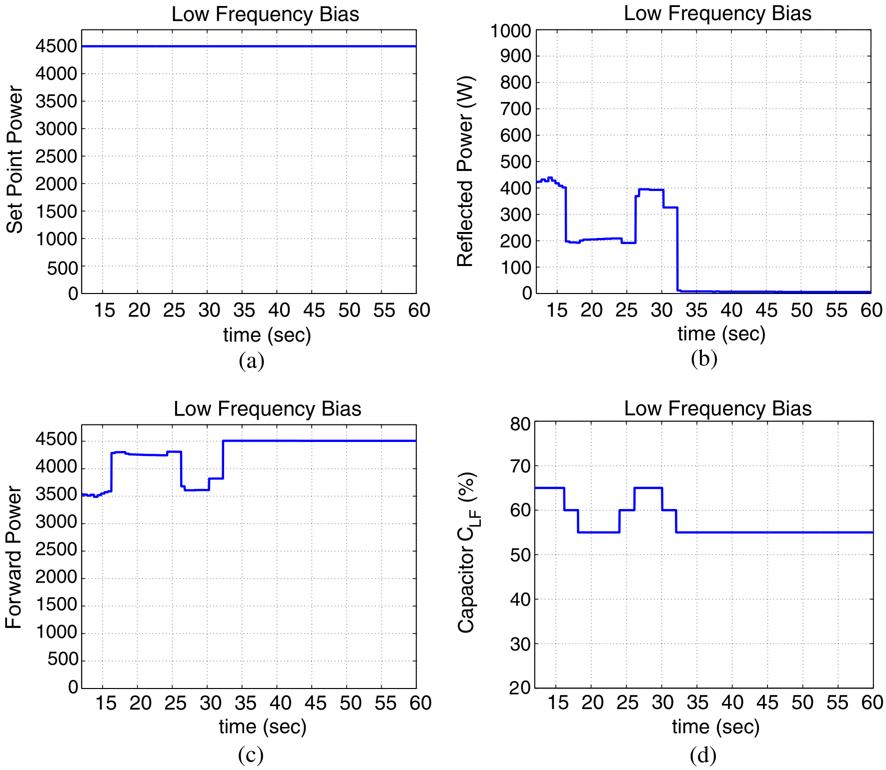


Fig. 7.12 Direct search based extremum seeking control for low frequency matching network: (**a**) set point of 4500 W; (**b**) reflected power; (**c**) forward power; (**d**) low frequency tune capacitor value

7.5 Conclusions

A study of impedance matching in semiconductor industry is presented in this chapter. From an engineering design point of view, the range of the series and shunt capacitors dictate the tuning space of the system. In other words, the capacitor values dictate the range of plasma load values that the system can support with minimum reflected power. This tuning space covers plasma loads for low and high pressure processes with a variety of gases of interest and for different power levels.

Due to practical industry needs, here we only focus on the matching network design via extremum seeking control. However, the same concept can be applied to frequency tuning. Compared with the capacitor matching network with generator frequency tuning problem in Sect. 7.4, one would suggest that NOESC should be used in the matching network problem of Sect. 7.3, as it consists of mechanically adjustable components and therefore tuning is done on the level of set-point generation. As for the generator frequency tuning problem, AOESC could also be used, since frequency tuning requires a very fast response time (down to a few microseconds). Moreover, setting the right presets for the series and shunt capacitors will

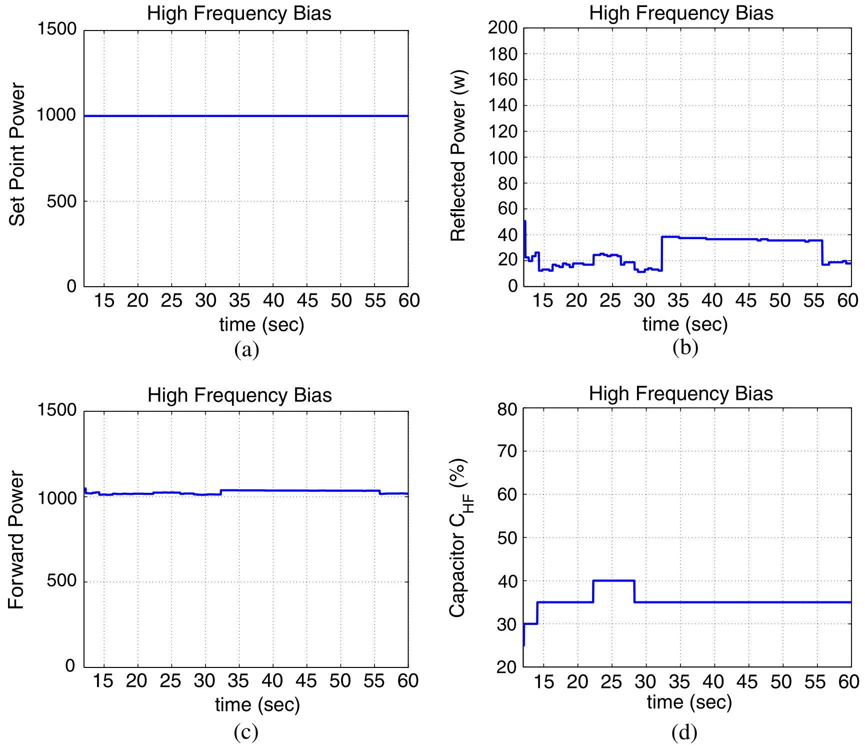


Fig. 7.13 Direct search based extremum seeking control for high frequency matching network: (**a**) set point of 1000 W; (**b**) reflected power; (**c**) forward power; (**d**) low frequency tune capacitor value

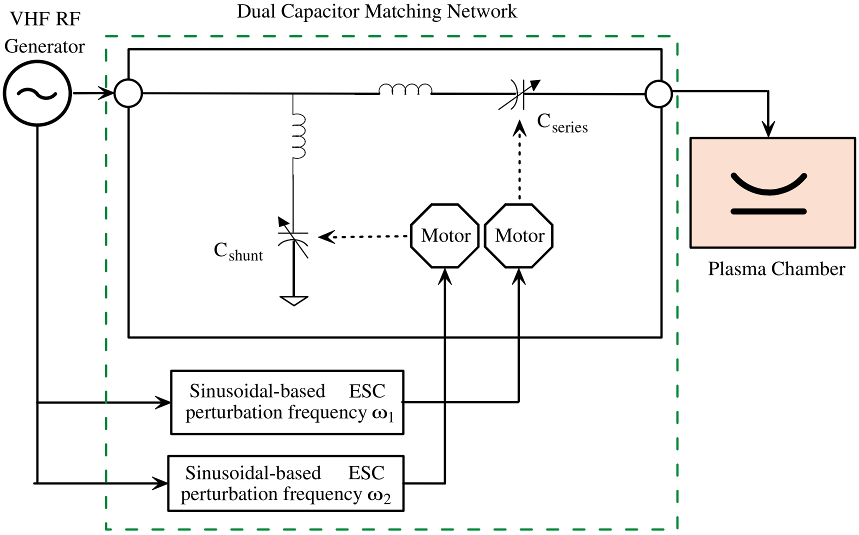


Fig. 7.14 The dual capacitor matching network

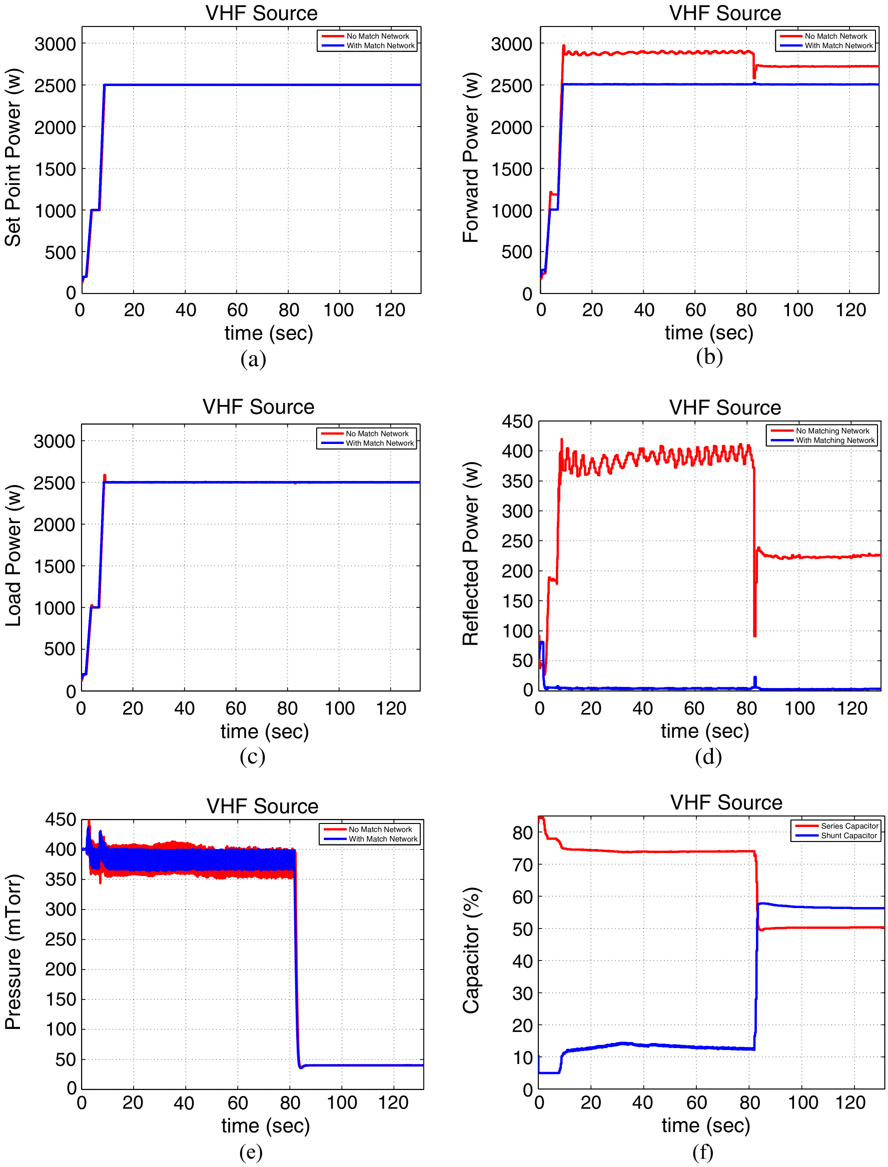


Fig. 7.15 Sinusoidal perturbation based extremum seeking control for very high frequency matching network: (**a**) set point; (**b**) forward power; (**c**) load power; (**d**) reflected power; (**e**) pressure; (**f**) tune capacitor values

ensure high voltage for plasma ignition along with low reflected power while sustaining the plasma. The capacitor matching network generally has a wide tuning range while the frequency tuning range is much narrower (as one does not want to have large frequency sweeps, which could perturb the plasma). In some applications, one could consider combining both frequency tuning and capacitor matching network to realize impedance tuning for a wide range of process conditions.

References

1. Advanced Energy Industries. http://www.advancedenergy.com/
2. Ariyur, K.B., Krstić, M.: Real-time Optimization by Extremum-seeking Control. Wiley-Interscience, Hoboken (2003)
3. Carter, D.: Optimized process performance using the paramount navigator power-delivery/match solution. White Paper, Advanced Energy Industries, Inc (2007)
4. Daihen Advanced Component. http://www.daihen-ac.com/
5. Liberman, M.A., Lichtenberg, A.J.: Principles of Plasma Discharges and Materials Processing. Wiley, New York (2005)
6. Maloratsky, L.G.: Couplers shrink hf/vhf/uhf designs. Microw. RF 93–96 (2000)
7. Maloratsky, L.G.: Understand the basics of microstrip directional couplers. Microw. RF 79–94 (2001)
8. Moritz, J.R., Sun, Y.: Frequency agile antenna tuning and matching. In: 8th International Conference on High-Frequency Radio Systems and Techniques (2000)
9. Parro, V.C., Pait, F.M.: An automatic impedance matching system for a multi point power delivery continuous industrial microwave oven. In: Proceedings of the 2004 IEEE International Conference on Control Applications, vol. 1, pp. 143–148 (2004)
10. Rotea, M.A.: Analysis of multivariable extremum seeking algorithms. In: Proceedings of the American Control Conference, vol. 1, pp. 433–437 (2000)
11. Scholl, R.A.: Forward and reflected powers, what do they mean? White Paper. Advanced Energy Industries, Inc (1998)
12. Seren Industrial Power Systems. http://www.serenips.com/
13. Sun, Y., Fidler, J.K.: High speed automatic antenna tuning units. In: Proceeding of IEE Conference on Antennas and Propagation (1995)
14. Sun, Y., Lau, W.K.: Automatic impedance matching using genetic algorithms. In: Proceeding of IEE Conference on Antennas and Propagation (1999)
15. van Bezooijen, A., de Jongh, M.A., van Straten, F., Mahmoudi, R., van Roermund, A.: Adaptive impedance-matching techniques for controlling l networks. IEEE Trans. Circuits Syst. I, Regul. Pap. **57**(2), 495–505 (2010)
16. Whatley, R.B., Zhou, Z., Melde, K.L.: Reconfigurable rf impedance tuner for match control in broadband wireless devices. IEEE Trans. Antennas Propag. **54**(2), 470–478 (2006)
17. Zhang, C., Ordóñez, R.: Non-gradient extremum seeking control of feedback linearizable systems with application to ABS design. In: Proceedings of the Conference Decision and Control, pp. 6666–6671 (2006)
18. Zhang, C., Shoji, S.F., Semenin, A., Ramaswamy, K., Cruse, J.P., Liao, B.: Methods and apparatus for tuning matching newtorks. United States Patent Application Serial No. 61/253,727 (2009)
19. Zhang, C., Wong, L., Ramaswamy, K., Cruse, J.P., Hanawa, H.: Plasma Reactor with RF generator and automatic impedance match with minimum reflected power-seeking control. United States Patent Application Serial No. 12/502,037 (2009)
20. Zhang, C., Shoji, S.F., Semenin, A., Ramaswamy, K., Cruse, J.P., Liao, B.: Methods and apparatus for tuning matching networks. United States Patent Application Serial No. 12/899,048, also submitted patent application in Taiwan and China (2010)

Chapter 8
Swarm Tracking

8.1 Introduction

The problem of coordination and control of autonomous vehicles, autonomous robots, or agents, has been receiving an extraordinary amount of attention during the past decade. The same is true of the study of the biological world and the application of derived principles to engineering designs. Instead of the traditional trajectory tracking problem, some researchers began to study coordinated tracking [33, 42]. Some tasks can be performed more efficiently by controlling a group of agents in a collaborative manner. Possible applications could range from autonomous robot assembly to unmanned air vehicles (UAVs) scout and counter insurgency. Compared to individuals, swarms, flocks, and schools can have remarkable group-level characteristics, which may allow them to perform complicated tasks efficiently.

As discussed earlier in Chap. 1, extremum seeking is first applied to source seeking in [48], and a recent overview of source seeking can be found in [25]. Detailed work on source seeking can be found in [4–6, 8, 9, 11, 13, 14, 20, 21, 28, 29, 39, 40, 49], where perturbation based design is the focus.

The swarm tracking problem consists of finding a coordinated control scheme for a group of agents, with the objective of making them achieve and maintain some given geometrical formation. At the same time, the agents need to seek a source of a scalar signal or track a moving target. Thus, there is a trade-off between maintaining formation and arriving at the final goal. Related work can be found in [3, 18, 19, 30]. Motivated by the work in [43] and [48], swarm tracking is achieved via artificial potentials and extremum seeking control in this chapter.

Artificial potential functions have been widely used for robot navigation and control, including multi-agent coordination [12, 27, 34, 35]. A potential function is created to contain the scalar signal to be tracked, as well as the interaction rules for the group of agents. By minimizing the potential function, one is able to achieve source seeking, formation control, obstacle avoidance and collision avoidance. Extremum seeking techniques [1] are used to design the controller for each agent, and three different extremum seeking control designs are studied here. Among these, gradient extremum seeking is the natural extension from the work in [43]. Perturbation based

C. Zhang, R. Ordóñez, *Extremum-Seeking Control and Applications*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2224-1_8, © Springer-Verlag London Limited 2012

extremum seeking has been studied for source seeking in [7, 10, 48, 49]. Numerical optimization based extremum seeking control [46] is also used here for source seeking.

We begin this chapter with a problem statement, introducing the model for each agent and the design block diagram in Sect. 8.2. Three different extremum seeking control designs are presented in Sects. 8.3, 8.4 and 8.5, respectively. Section 8.6 presents a specific application of the swarming theory of Sect. 8.5 to the problem of localizing radar leakage points via a mobile sensor network. Finally, Sect. 8.7 concludes the chapter.

8.2 Problem Statement

Consider a multi-agent system (i.e., a swarm) consisting of N individuals in n-dimensional Euclidean space. We assume synchronous motion and no time delays. Let $x^i \in \mathbb{R}^n$ denote a column vector in n-dimensional Euclidean space, whose meaning is the position of an individual agent i. Furthermore, let $x^\top = [x_1^\top, \ldots, x_N^\top] \in \mathbb{R}^{n \times N}$. To begin with, assume that the ith agent's motion is governed by the following point-mass kinematic model,

$$\dot{x}^i = u^i, \tag{8.1}$$

where $u^i \in \mathbb{R}^n$ is the control input for the ith agent. We simplify the dynamics to be kinematic in order to focus on the coordinated control design for swarm tracking. Later we also consider a double integrator point-mass model. For other more realistic models such as the unicycle, techniques like phase lead compensator [48], sliding mode control [24] or trajectory tracking can be used to extend the work here.

Furthermore, we assume there is a scalar signal $J_t(x)$ to be tracked by the swarming agents, which has an unknown isolated minimum at $x_t \in \mathbb{R}^n$. The strength of the scalar signal can be measured by the agent. Such signal could be a field signal generated by a mobile source x_t, or an artificial potential field if tracking a moving target and one knows its position x_t, or the relative distance between agent and target. Our purpose is to design a control law for each agent such that we can achieve swarm tracking, i.e., source seeking, formation control, collision avoidance and obstacle avoidance.

We consider a potential function composed of several components. First, the inter-connection component puts a constraint on the agent, based on its neighbors' positions, in order to maintain a group structure. This part includes functions of the relative distance between each pair of neighbors. In addition, a tracking component containing the scalar signal to be tracked is added in order to direct the group's behavior for source seeking. This tracking component could be an artificial potential function given the knowledge of target position, or the concentration of a chemical source. It could even be an electromagnetic (as in Sect. 8.6), an acoustic or a thermal signal. The specific form of the potential function is defined according to the desired geometric formation. The choice of a potential function is important because different potentials might result in different performance even with the same control algorithm. In particular, existence of multiple local minima in the potential function results in only being able to guarantee local convergence to the desired formation. Nevertheless, we show that by appropriate choice of the potential function one can always guarantee that eventually the target will be surrounded or "enclosed" by the tracking agents.

First, consider a potential function for agent to target interaction given by

$$J_{at}(x, x_t) = \sum_{i=1}^{N} J_{it}\big(\|x^i - x_t\|\big), \tag{8.2}$$

where J_{it} is constructed based on the scalar signal J_t, and it is assumed to have a minimum at $\|x^i - x_t\| = \delta_{it}$ of J_{it}. In other words, we require each agent to track x_t by a prescribed distance of δ_{it}. One can calculate such potential function based on the $J_t(x)$ measurement even if x_t is not known. Moreover, we do not want δ_{it} to be zero since it will result in collision of the agent with the target. Also if the agent is modeled as a unicycle with non-collocated sensor [23, 26], we can have the sensor position of each agent tracking the target, while the agent itself is at a distance r away from the target, where r is the distance between the agent and sensor footprint. A similar situation can happen in the case of air vehicles tracking a ground target, where we might want the sensor footprint of each air vehicle focusing on the target but avoiding collision.

Second, consider the potential function for agent to agent interaction, given by

$$J_{aa}(x) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} J_{ij}\big(\|x^i - x^j\|\big), \tag{8.3}$$

where $J_{ij}(\|x^i - x^j\|)$ is the potential between the ith and the jth agent, whose purpose is to achieve a balance between attraction and repulsion [17]. Such potentials can be obtained if one knows the relative distance between the agents, or by measuring a possible field distribution generated by the agent.

Now, we can put the swarm tracking problem into the framework of extremum seeking control design. Slightly abusing notation, let

$$\begin{aligned} y &= J(x, x_t) \\ &= J\big(\|x^i - x_t\|, \|x^i - x^j\|\big), \quad \text{for } 1 \le i, j \le N \\ &= K_{at} J_{at}(x, x_t) + K_{aa} J_{aa}(x) \\ &= K_{at} \sum_{i=1}^{N} J_{it}\big(\|x^i - x_t\|\big) + K_{aa} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} J_{ij}\big(\|x^i - x^j\|\big), \end{aligned} \tag{8.4}$$

be the performance function of the system (8.1) for $i = 1, \dots, N$, where K_{at}, K_{aa} are the weights of the potential components. Our objective is to make the entire

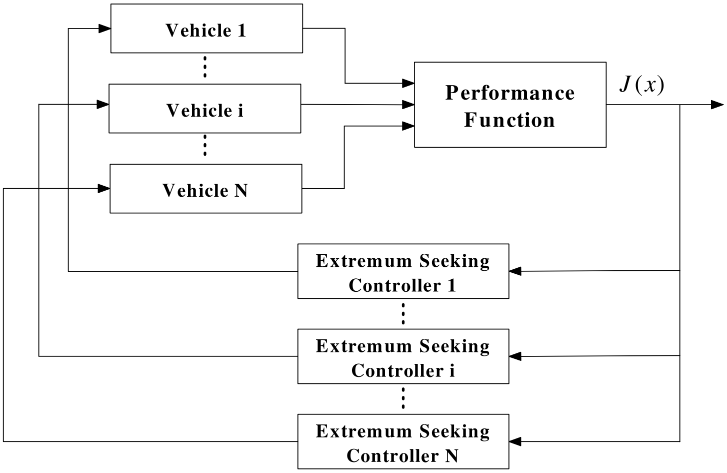

Vehicle 1
Vehicle i
Vehicle N
Performance Function
$J(x)$
Extremum Seeking Controller 1
Extremum Seeking Controller i
Extremum Seeking Controller N

Fig. 8.1 Swarm tracking for N agents/vehicles

group aggregate around the target x_t and track along it, possibly in a specific formation regardless of the target's movement. A block diagram can be found in Fig. 8.1.

Keeping in mind that we have assumed the potential function will have a minimum at $\|x^i - x_t\| = \delta_{it}$ and $\|x^i - x^j\| = \delta_{ij}$, if the extremum seeking controller for each agent can minimize the performance function (8.4), then we can achieve our objective of source seeking, formation control and collision avoidance (this objective is achieved by adding a repulsive potential between the agents and obstacle into (8.4)). This control design will be decentralized if each agent has its own performance function, which will be discussed in Sects. 8.4, 8.5 and 8.6.

8.3 Gradient Based Extremum Seeking Control Design

8.3.1 Analysis

Based on [43], we present the gradient based extremum seeking control design for each agent. First, we assume that the following conditions hold for the potential function J:

Assumption A1 *For $i = 1, \ldots, N$, there exist functions $h^{it} : \mathbb{R}^+ \to \mathbb{R}$ such that*

$$\nabla_x J_{it}\big(\|x\|\big) = x h^{it}\big(\|x\|\big).$$

Assumption A2 *There exist unique distances δ_{it} at which we have*

$$h^{it}(\delta_{it}) = 0.$$

Assumption A3 *The potentials* $J_{ij}(\|x^i - x^j\|)$ *are symmetric and satisfy*

$$\nabla_{x^i} J_{ij}\big(\|x^i - x^j\|\big) = -\nabla_{x^j} J_{ij}\big(\|x^i - x^j\|\big).$$

Assumption A4 *For* $1 \le i, j \le N$ *there exist functions* $g_{ar}^{ij} : \mathbb{R}^+ \to \mathbb{R}$ *such that*

$$\nabla_x J_{ij}\big(\|x\|\big) = x g_{ar}^{ij}\big(\|x\|\big).$$

Assumption A5 *There exist unique distances* δ_{ij} *at which we have*

$$g_{ar}^{ij}\big(\|x\|\big) \begin{cases} > 0, & \|x\| > \delta_{ij}, \\ = 0, & \|x\| = \delta_{ij}, \\ < 0, & \|x\| < \delta_{ij}. \end{cases}$$

Potential functions satisfying Assumptions A3 to A5 are odd functions that are attractive on distances $\|x\| > \delta_{ij}$ and repulsive on distances $\|x\| < \delta_{ij}$. The term $g_{ar}^{ij}(\|x\|)$ determines the attraction-repulsion relationship between the individuals and usually is of the form

$$g_{ar}^{ij}\big(\|x\|\big) = g_a^{ij}\big(\|x\|\big) + g_r^{ij}\big(\|x\|\big),$$

where $g_a^{ij}(\|x\|)$ represents the attraction and $g_r^{ij}(\|x\|)$ represents the repulsion. The distance δ_{ij} is the equilibrium distance at which the attraction and the repulsion balance. That is, the potential function $J_{ij}(\|x^i - x^j\|)$ has a minimum at $\|x^i - x^j\| = \delta_{ij}$. Later, we will see that if we can construct a potential function J_{aa} encoding a desired formation of the agents, then the formation control will be realized if such potential function is minimized. Similarly, we can set up a repulsive potential function J_{ao} between obstacle and agents, where in this case the potential function is inverse proportional to the distance between obstacle and agents. By doing so, we can achieve obstacle avoidance by minimizing J_{ao}.

We now study some consequences of Assumptions A1 through A5, similar to the analysis in [43]. From (8.4) we obtain

$$\begin{aligned} \nabla_{x_i} J(x, x_t) = {} & K_{at}(x_i - x_t) h^{it}\big(\|x_i - x_t\|\big) \\ & + K_{aa} \sum_{j=1, j\neq i}^{N} (x_i - x_j) g_{ar}^{ij}\big(\|x_i - x_j\|\big) \end{aligned} \tag{8.5}$$

and

$$\nabla_{x_t} J(x, x_t) = -K_{at} \sum_{i=1}^{N} (x_i - x_t) h^{it}\big(\|x_i - x_t\|\big). \tag{8.6}$$

By observing the equalities in (8.5) and (8.6), notice that the equality in (8.6) can be rewritten as

$$\begin{aligned}\nabla_{x_t} J(x, x_t) = &-\sum_{i=1}^{N} \nabla_{x_i} J(x, x_t) \\ &+ K_{aa} \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} (x_i - x_j) g_{ar}^{ij}\big(\|x_i - x_j\|\big).\end{aligned} \tag{8.7}$$

Moreover, since we have

$$\sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} (x_i - x_j) g_{ar}^{ij}\big(\|x_i - x_j\|\big) = 0, \tag{8.8}$$

which follows from Assumption A3, we obtain

$$\nabla_{x_t} J(x, x_t) = -\sum_{i=1}^{N} \nabla_{x_i} J(x, x_t). \tag{8.9}$$

Now, without loss of generality, we can choose a Lyapunov candidate as $V = J(x, x_t)$. Then, it follows that

$$\begin{aligned}\dot{V} &= \sum_{i=1}^{N} \big[\nabla_{x^i} J(x, x_t)\big]^\top \dot{x}^i + \big[\nabla_{x^t} J(x, x_t)\big]^\top \dot{x}_t \\ &= \sum_{i=1}^{N} \big[\nabla_{x^i} J(x, x_t)\big]^\top \big(u^i - \dot{x}_t\big),\end{aligned}$$

so for the ith agent, let

$$u^i = \dot{x}_t - k_i \nabla_{x^i} J(x, x_t), \quad \text{for } k_i > 0. \tag{8.10}$$

Then we obtain

$$\dot{V} = -\sum_{i=1}^{N} k_i \big\|\nabla_{x^i} J(x, x_t)\big\|^2 \leq 0.$$

By the LaSalle–Yoshizawa theorem [38], we can further conclude that the trajectory of x asymptotically converges to values for which

$$\big\|\nabla_{x^i} J(x, x_t)\big\| = 0, \tag{8.11}$$

$$\big\|\nabla_{x^t} J(x, x_t)\big\| = 0. \tag{8.12}$$

Then, for values of (x, x_t) satisfying (8.11) and (8.12) we have, from (8.6),

$$-K_{at}\sum_{i=1}^{N}(x_i - x_t)h^{it}\big(\|x_i - x_t\|\big) = 0.$$

Rearranging this equation, we obtain

$$\sum_{i=1}^{N} x_i h^{it}\big(\|x_i - x_t\|\big) = x_t \sum_{i=1}^{N} h^{it}\big(\|x_i - x_t\|\big), \tag{8.13}$$

which is guaranteed to be achieved as $t \to \infty$. Equation (8.13) provides a relation of the position of the target to the position of the agents at equilibrium and allows the designer to choose appropriately the functions $h^{it}(\|x_i - x_t\|)$ (that is, the tracking part of the potential function).

First, since $\|x_i - x_t\| = \delta_{it}$ for all i, note that at the desired formation we have $h^{it}(\|x_i - x_t\|) = 0$ for all i, and (8.13) is satisfied. In this case, agents catch up with the target and compose the expected formation with respect to it.

Second, note that if $h^{it}(\|x_i - x_t\|)$ are chosen such that

$$\sum_{i=1}^{N} h^{it}\big(\|x_i - x_t\|\big) = 0$$

can occur (excluding the case at the desired formation), then the position of the target x_t cannot be specified (meaning that it could be anywhere in the state space). To avoid this situation one can choose $h^{it}(\|z\|) > 0$ for all z except $\|z\| = \delta_{it}$. Then, assuming that

$$\sum_{i=1}^{N} h^{it}\big(\|x_i - x_t\|\big) \neq 0,$$

we obtain

$$x_t = \frac{\sum_{i=1}^{N} x_i h^{it}(\|x_i - x_t\|)}{\sum_{i=1}^{N} h^{it}(\|x_i - x_t\|)}.$$

Defining

$$\eta_i = \frac{h^{it}(\|x_i - x_t\|)}{\sum_{i=1}^{N} h^{it}(\|x_i - x_t\|)}, \quad i = 1, \ldots, N,$$

we can write

$$x_t = \sum_{i=1}^{N} \eta_i x_i.$$

With the choice of $h^{it}(\|z\|) \geq 0$ for all z, we see that $0 \leq \eta_i \leq 1$ for all i and $\sum_{i=1}^{N} \eta_i = 1$, implying that as $t \to \infty$ we will have

$$x_t \to conv\{x_1, x_2, \ldots, x_N\},$$

where $conv\{x_1, x_2, \ldots, x_N\}$ is the convex hull of the positions of the agents. In other words, by choosing $h^{it}(\|z\|)$ as above one can guarantee that as $t \to \infty$, the agents will "surround" or "enclose" the target.

However, assuming the measurement of $\dot{x}_t$ in the control signal (8.10) is a strong assumption, since usually it is not possible for one to know the velocity of the source. This is especially true since one often does not even know the source's position, x_t. Thus, we will only assume that $\|\dot{x}_t\| \leq \gamma_t$ for some known $\gamma_t > 0$. Note that this constitutes a more realistic assumption, since any moving target has a bounded velocity. With this assumption, let $k_i > 0$, $\beta_i > \gamma_t$, and choose the controller as

$$u^i = -k_i \nabla_{x^i} J(x, x_t) - \beta_i \operatorname{sgn}\big(\nabla_{x^i} J(x, x_t)\big), \tag{8.14}$$

where sgn(·) is the signum function. Then,

$$\begin{aligned} \dot{V} &= \sum_{i=1}^{N} \Big(-k_i \big\|\nabla_{x^i} J(x, x_t)\big\|^2 - \beta_i \big\|\nabla_{x^i} J(x, x_t)\big\| - \big[\nabla_{x^i} J(x, x_t)\big]^\top \dot{x}_t\Big) \\ &\leq -\sum_{i=1}^{N} k_i \big\|\nabla_{x^i} J(x, x_t)\big\|^2 \leq 0. \end{aligned} \tag{8.15}$$

We can further relax the controller design to

$$u^i = -(k_i + \beta_i) \operatorname{sgn}\big(\nabla_{x^i} J(x, x_t)\big). \tag{8.16}$$

Then, we obtain

$$\begin{aligned} \dot{V} &= \sum_{i=1}^{N} \Big(- (k_i + \beta_i) \big\|\nabla_{x^i} J(x, x_t)\big\| - \big[\nabla_{x^i} J(x, x_t)\big]^\top \dot{x}_t\Big) \\ &\leq -\sum_{i=1}^{N} k_i \big\|\nabla_{x^i} J(x, x_t)\big\| \leq 0. \end{aligned} \tag{8.17}$$

Therefore, the controllers (8.14) and (8.16) again result in the agent positions x^i asymptotically enclosing the target x_t from δ_{it} distance away. This result requires the knowledge of the gradient of the scalar signal, or at least the sign of the gradient and a bound on the target speed. Then, with the help of a switching term, the controllers guarantee asymptotic tracking of the target. Intuitively, the signum function allows for the detection of the changes in the direction of the motion of the target and helps redirect the agent in that direction. All the three controllers (8.10), (8.14) and (8.16) belong to the gradient based extremum seeking controller type.

8.3.2 Kinematic Point-Mass Model Simulations

For simulation purposes, we consider a two-dimensional case where $n = 2$ and three agents, so $N = 3$. The scalar signal is postulated as a quadratic function:

$$J_t(x, x_t) = \|x - x_t\|^2, \tag{8.18}$$

where the dynamics of the target x_t satisfy

$$\begin{aligned} \dot{x}_{t_1} &= 0.25, \\ \dot{x}_{t_2} &= \sin(0.25t). \end{aligned} \tag{8.19}$$

Then we construct the potential function

$$J_{at}(x, x_t) = \sum_{i=1}^{3} J_{it}\big(\|x^i - x_t\|\big),$$

where

$$J_{it}\big(\|x^i - x_t\|\big) = \frac{1}{2}\big(\|x^i - x_t\|^2 - \delta_{it}^2\big)^2$$

for $i = 1, 2, 3$. We can further verify that

$$\nabla_{x^i} J_{it}\big(\|x^i - x_t\|\big) = 2\big(\|x^i - x_t\|^2 - \delta_{it}^2\big)\big(x^i - x_t\big) = 0$$

if $\|x^i - x_t\| = \delta_{it}$ as assumed in Assumption A2. Moreover, we choose the potential function between agents as

$$J_{aa}(x) = \sum_{i=1}^{2} \sum_{j=2}^{3} J_{ij}\big(\|x^i - x^j\|\big),$$

where

$$J_{ij}\big(\|x^i - x^j\|\big) = \frac{1}{2}\big(\|x^i - x^j\|^2 - \delta_{ij}^2\big)^2$$

for $i = 1, 2, 3$. And we can also verify

$$\nabla_{x^i} J_{ij}\big(\|x^i - x^j\|\big) = 2\big(\|x^i - x^j\|^2 - \delta_{ij}^2\big)\big(x^i - x^j\big)$$

and

$$\nabla_{x^j} J_{ij}\big(\|x^i - x^j\|\big) = -2\big(\|x^i - x^j\|^2 - \delta_{ij}^2\big)\big(x^i - x^j\big),$$

which satisfies Assumption A3. Such design of potential function J_{aa} prescribes a triangular formation for the agents to form with each lateral length equal to δ_{ij}.

Now, the performance function becomes

$$y = J(x, x_t) = K_{at} J_{at}\big(x^1, x^2, x^3, x_t\big) + K_{aa} J_{aa}\big(x^1, x^2, x^3\big), \tag{8.20}$$

where the weights K_{at}, K_{aa} are important in balancing the priority in source seeking and formation keeping. In the simulation, we choose $\delta_{it} = 1$, $\delta_{ij} = \sqrt{3}$ for $i, j = 1, 2, 3$, $K_{at} = 1$, $K_{aa} = 0.1$, and $k_i = 10$, $\beta_i = 2$ for $i = 1, 2, 3$. We apply the controller (8.14) to each agent, all of which start randomly inside a ball centered at $(-1, -1)$ with radius 0.01, and the target is started from position $[0, 0]^\top$. The simulation time interval is $[0, 50]$ time units. The results can be found in Fig. 8.2. The movements of the agents and the target over the entire simulation can be found in Fig. 8.2(a), where source seeking, formation control and collision avoidance are successfully achieved This figure shows "snapshots" of the formation at various times between beginning and end of the simulation. The final triangular formation (that is, at $t = 50$ time units) can be found in Fig. 8.2(b), where the stars denote the center trajectory of all three agents following exactly the target trajectory, and the circles denote the agent positions surrounding the target. Also, the performance functions are minimized in a very short time period as seen in Fig. 8.2(c), which shows the values of the performance functions over time.

8.3.3 Dynamic Point-Mass Model Simulations

Consider also a point-mass with second order dynamics, that is,

$$\ddot{x}^i = u^i, \tag{8.21}$$

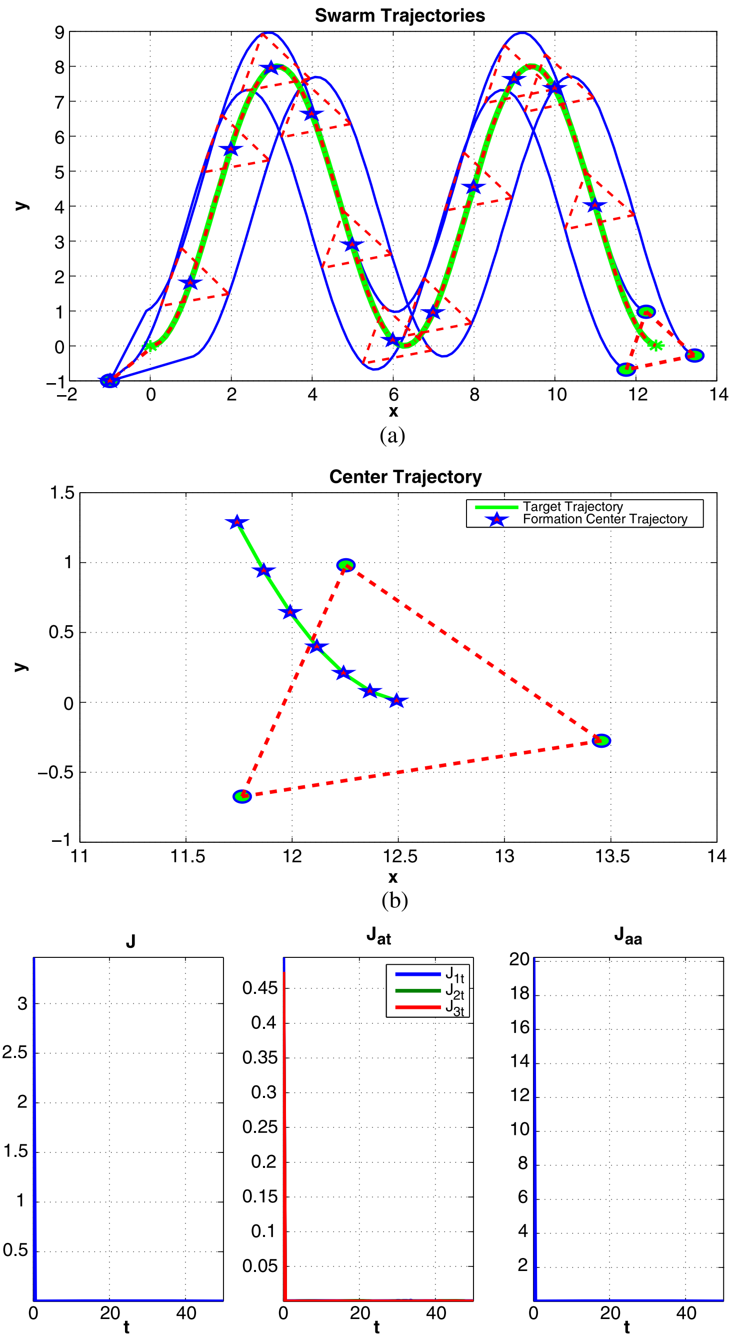
instead the kinematic equation (8.1). We can follow the controller design for a single vehicle as in [16], where a sliding mode is used to force the sliding mode occur on the gradient flow. The simulation results can be found in Fig. 8.3. The results shown here follow the same format as those in Fig. 8.2.

8.4 Perturbation Based Extremum Seeking Control Design

We achieved very good simulation results by using the gradient based extremum seeking control in Sect. 8.3. However, the controllers used there rely on the knowledge of the gradient of the performance function, which is a very strong assumption. Moreover, for complicated obstacles near the agent, it is difficult to calculate the gradient even if one knows the analytical form of the potential function and the relative distance between the agent and the obstacle. In this section, we apply perturbation based extremum seeking control to each agent [48].

8.4.1 Analysis

Let $x^i = [x_p^i, y_p^i]^\top$ denote the ith agent's position in two-dimensional space, with kinematic equations $\dot{x}_p^i = u_x^i$, $\dot{y}_p^i = u_y^i$. The controller is designed as

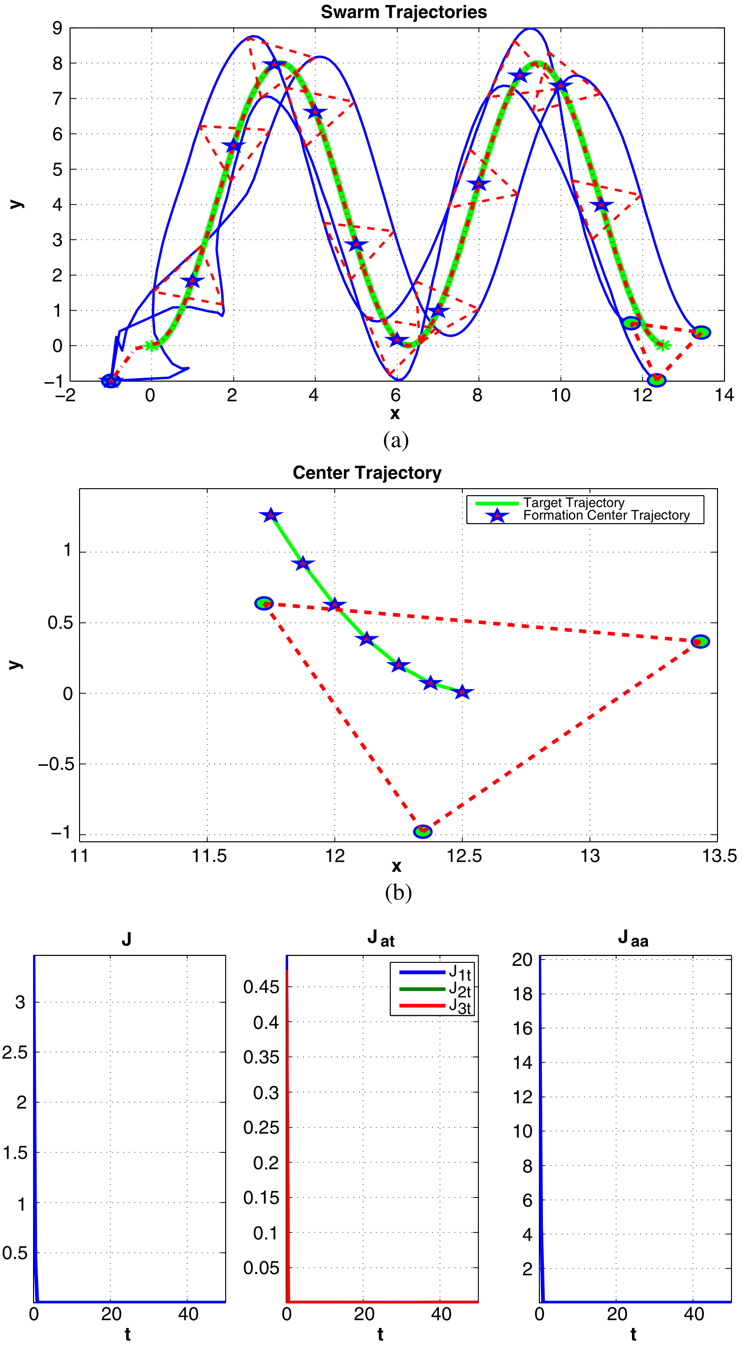


(a)

(b)

(c)

Fig. 8.2 Swarm tracking via gradient based extremum seeking control, kinematic point-mass: (**a**) trajectories of the swarm agents; (**b**) final formation; (**c**) potential functions



(a)

(b)

(c)

Fig. 8.3 Swarm source seeking via gradient based extremum seeking control, dynamic point-mass: (**a**) trajectories of the swarm vehicles; (**b**) final formation; (**c**) potential functions

$$u_x^i = c_x^i \sin(\omega_i t) \frac{s}{s+h_i}[J] + \alpha_i \omega_i \cos(\omega_i t), \tag{8.22}$$

$$u_y^i = -c_y^i \cos(\omega_i t) \frac{s}{s+h_i}[J] + \alpha_i \omega_i \sin(\omega_i t), \tag{8.23}$$

where the notation $\frac{s}{s+h_i}[J]$ is used to represent the output of the block whose transfer function is $\frac{s}{s+h_i}$ and whose input is J. The constants are chosen as c_x^i, c_y^i, h_i, $\alpha_i > 0$ for the ith agent, and the perturbation frequencies are $\omega_i \neq \omega_j$ for $i \neq j$. It is the use of the high pass filter and perturbation signal that extracts the gradient information from the performance function measurement.

Now, consider the performance function (8.20): the first agent is only able to influence the potential functions J_{1t}, J_{12} and J_{13}. Therefore, instead of giving each agent the same performance function, we replace J in (8.22) with J_i for the ith agent, where

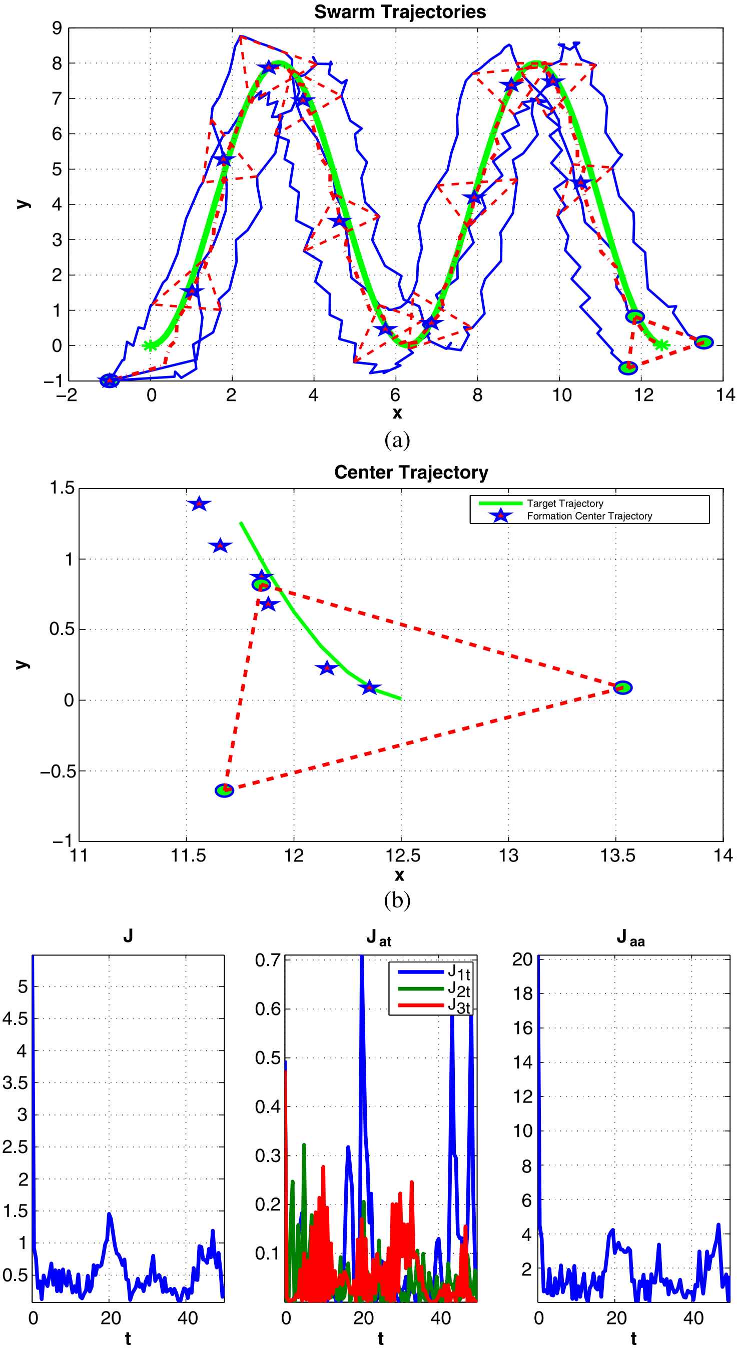
$$\begin{aligned} J_i\big(x^i, x_t\big) &= K_{at} J_{it}\big(\big\|x^i - x_t\big\|\big) + K_{aa} \sum_{j=1, j\neq i}^{3} J_{ij}\big(\big\|x^i - x^j\big\|\big) \\ &= \frac{K_{at}}{2}\big(\big\|x^i - x_t\big\|^2 - \delta_{it}^2\big)^2 + \frac{K_{aa}}{2} \sum_{j=1, j\neq i}^{3} J_{ij}\big(\big\|x^i - x^j\big\|^2 - \delta_{ij}^2\big). \end{aligned} \tag{8.24}$$

8.4.2 Kinematic Point-Mass Model Simulations

The simulation results can be found in Fig. 8.4, where the initial conditions are the same as in Sect. 8.3 and the format is the same as in Fig. 8.2. The parameters of the perturbation based extremum seeking control simulation are selected as $\omega_1 = 200$, $\omega_2 = 215$, $\omega_3 = 230$, $\alpha_i = 0.15$, and $c_x^i = c_y^i = 25$, $i = 1, 2, 3$. The weights of the potential function are chosen as $K_{at} = 1$ and $K_{aa} = 0.2$. The performance function is reduced to a neighborhood of the minimum, as we are tracking a moving target and no exact gradient information is available. The small peaks in the performance seen in Fig. 8.4(c) correspond to the peaks and valleys of the sinusoidal trajectory of the target, where the agent suddenly changes direction and increases the error of the gradient estimation. It is also important to notice that three different perturbation frequencies are needed for three agents in a two-dimensional space, and this choice of distinct frequencies is suggested from the stability analysis of multiple parameter perturbation based extremum seeking control in [36].

8.4.3 Obstacle Avoidance

The main advantage of the perturbation based extremum seeking control is that no gradient information and absolute position measurements are needed. The latter



(a)

(b)

(c)

Fig. 8.4 Swarm tracking via perturbation based extremum seeking control, kinematic point-mass: (**a**) trajectories of the swarm agents; (**b**) final formation; (**c**) potential functions

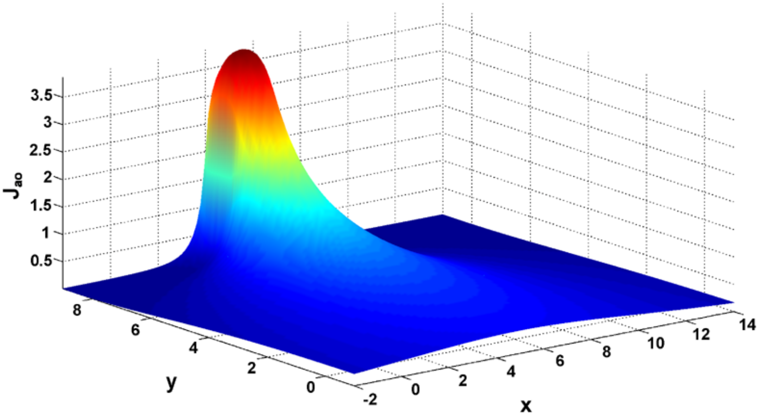


Fig. 8.5 Potential function of a rectangular obstacle

means no GPS or self-localization is required, as long as one can obtain the potentials. The former property is very appealing in incorporating new constraints or design requirements through potential functions. For instance, we can easily incorporate obstacle avoidance. For example, assume there is a rectangular obstacle with its coordinates given as $[o_1, o_2] \times [o_3, o_4]$, where $[o_1, o_2]$ is the x-axis coverage and $[o_3, o_4]$ is the y-axis coverage.

We can calculate the potential between the agent $x^i = [x_p^i, y_p^i]^\top$ and the rectangular obstacle with

$$J_{io}\bigl(x^i, o\bigr) = \Biggl| x_{o_1} \log\biggl(\frac{(x_{o_1}^2 + y_{o_1}^2)^{0.5} + y_{o_1}}{(x_{o_1}^2 + y_{o_1}^2)^{0.5} - y_{o_1}}\biggr) - x_{o_1} \log\biggl(\frac{(x_{o_1}^2 + y_{o_2}^2)^{0.5} + y_{o_2}}{(x_{o_1}^2 + y_{o_2}^2)^{0.5} - y_{o_2}}\biggr) + x_{o_2} \log\biggl(\frac{(x_{o_2}^2 + y_{o_2}^2)^{0.5} + y_{o_2}}{(x_{o_2}^2 + y_{o_2}^2)^{0.5} - y_{o_2}}\biggr) - x_{o_2} \log\biggl(\frac{(x_{o_2}^2 + y_{o_1}^2)^{0.5} + y_{o_1}}{(x_{o_2}^2 + y_{o_1}^2)^{0.5} - y_{o_1}}\biggr)\Biggr|,$$

where $o = [o_1, o_2, o_3, o_4]$, $x_{o_1} = o_1 - x_p^i$, $x_{o_2} = o_2 - x_p^i$, $y_{o_1} = o_3 - y_p^i$ and $y_{o_2} = o_4 - y_p^i$. As an illustration, a rectangular obstacle ranging from [4, 6] in the x-axis and [8, 9] in the y-axis can be found in Fig. 8.5.

Let

$$J_{ao}(x, o) = J_{1o}\bigl(x^1, o\bigr) + J_{2o}\bigl(x^2, o\bigr) + J_{3o}\bigl(x^3, o\bigr).$$

Then, the potential function (8.20) is augmented as

$$\begin{aligned} y &= J\bigl(x, x^t\bigr) \\ &= K_{at} J_{at}\bigl(x^1, x^2, x^3, x^t\bigr) + K_{aa} J_{aa}\bigl(x^1, x^2, x^3\bigr) + K_{ao} J_{ao}\bigl(x^1, x^2, x^3\bigr), \end{aligned}$$

where K_{ao} is the weight of the potential component generated by the obstacle. In general, we do not know where the obstacle is; however, if the agent is equipped for instance with a laser range finder, an infrared sensor or a visual sensor, among other possibilities, the relative distance to the obstacle can be found and it becomes possible to calculate a repulsive force similar to J_{io}. In such cases, it is very difficult or impossible to calculate the gradient of the potential function J_{io} and apply the gradient based extremum seeking control. However, we can easily incorporate such repulsive force into the performance function and apply perturbation based extremum seeking control, which is non-gradient based. Thus, we change the performance function J_i for the ith agent to

$$J_i\left(x^i, x_t, o\right) = K_{at} J_{it}\left(\left\|x^i - x_t\right\|\right) + K_{aa} \sum_{j=1, j\neq i}^{3} J_{ij}\left(\left\|x^i - x^j\right\|\right) + K_{ao} J_{io}\left(x^i, o\right).$$

Now the simulation results by including the obstacle $o = [4, 6, 8, 9]$ can be found in Fig. 8.6, where $k_{ao} = 0.5$. The obstacle is successfully avoided as seen in Fig. 8.6(b), and now a narrow peak is observed in the potentials when the obstacle is first encountered, as shown in Fig. 8.6(c). Here, the same presentation format as in Fig. 8.2 is used, except that Fig. 8.6(b) shows the formation at the time corresponding to its closest approach to the obstacle.

8.4.4 Dynamic Point-Mass Model Simulations

In the case when the vehicle is modeled by the dynamic point mass in (8.21), we can use the perturbation based extremum seeking design in Chap. 3 (the reader is also referred to [48] for more details).

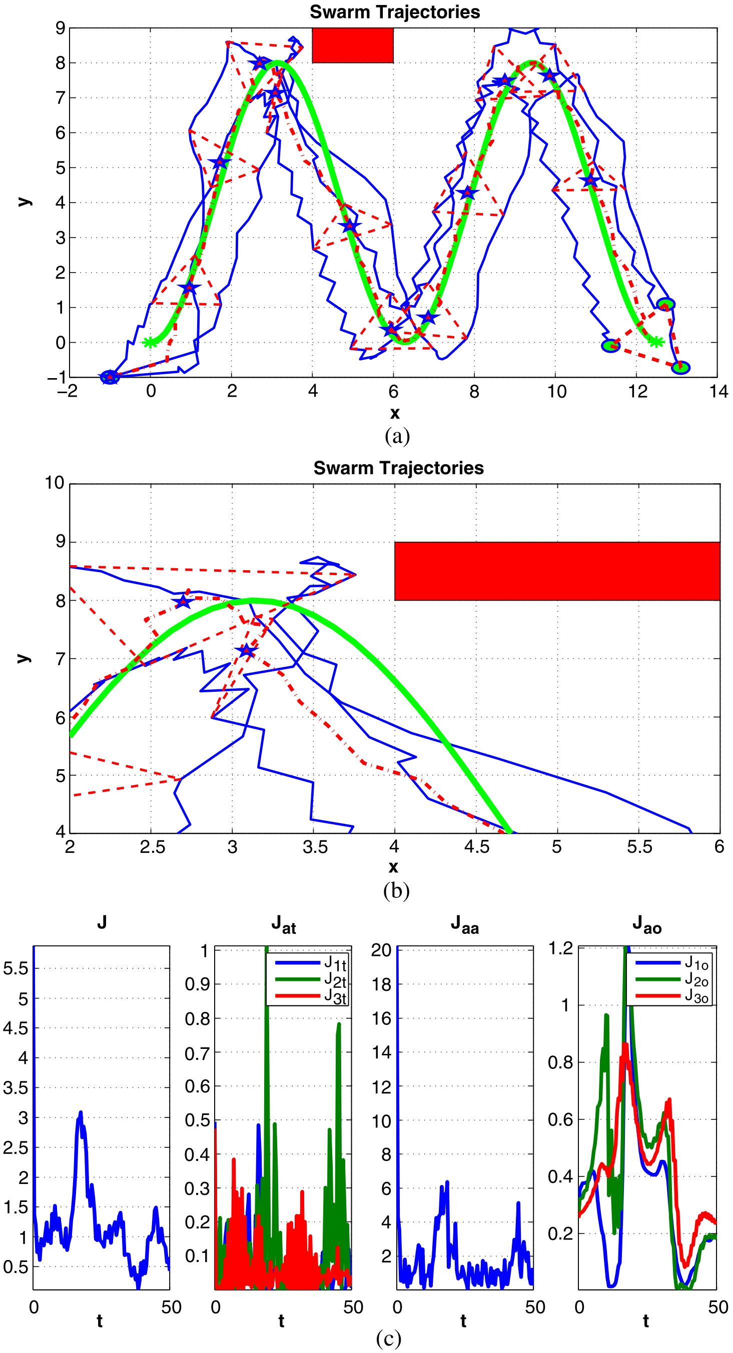
The simulation results can be found in Fig. 8.7, where the settings of the potential functions and initial conditions are the same as above and the presentation format is the same as in Fig. 8.2. The parameters of the perturbation based extremum seeking controller are $\omega_1 = 80$, $\omega_2 = 95$, $\omega_3 = 110$, $\alpha_i = 0.15$, and $c_{x_i} = c_{y_i} = 35$, $i = 1, 2, 3$. The zeros of the PD compensator are $z_{x_i} = -2$, the poles are $p_{x_i} = -5$ and the gains are $k_{x_i} = 1$ for $i = 1, 2, 3$. The weights of the potential function (8.24) are $K_{at} = 1$ and $K_{aa} = 0.2$. The simulation results for obstacle avoidance can be found in Fig. 8.8, where we add $K_{ao} = 0.5$.

8.5 Numerical Optimization Based Extremum Seeking Control Design

As the agent is modeled by (8.1), we rewrite the model as

$$\dot{x}^i = Ax^i + Bu^i,$$

where x^i, $u^i \in \mathbb{R}^n$, $A = 0$ and $B = [1, 1, \ldots, 1]^\top$. Therefore, we can apply numerical optimization based extremum seeking control [46], as discussed in Chaps. 4

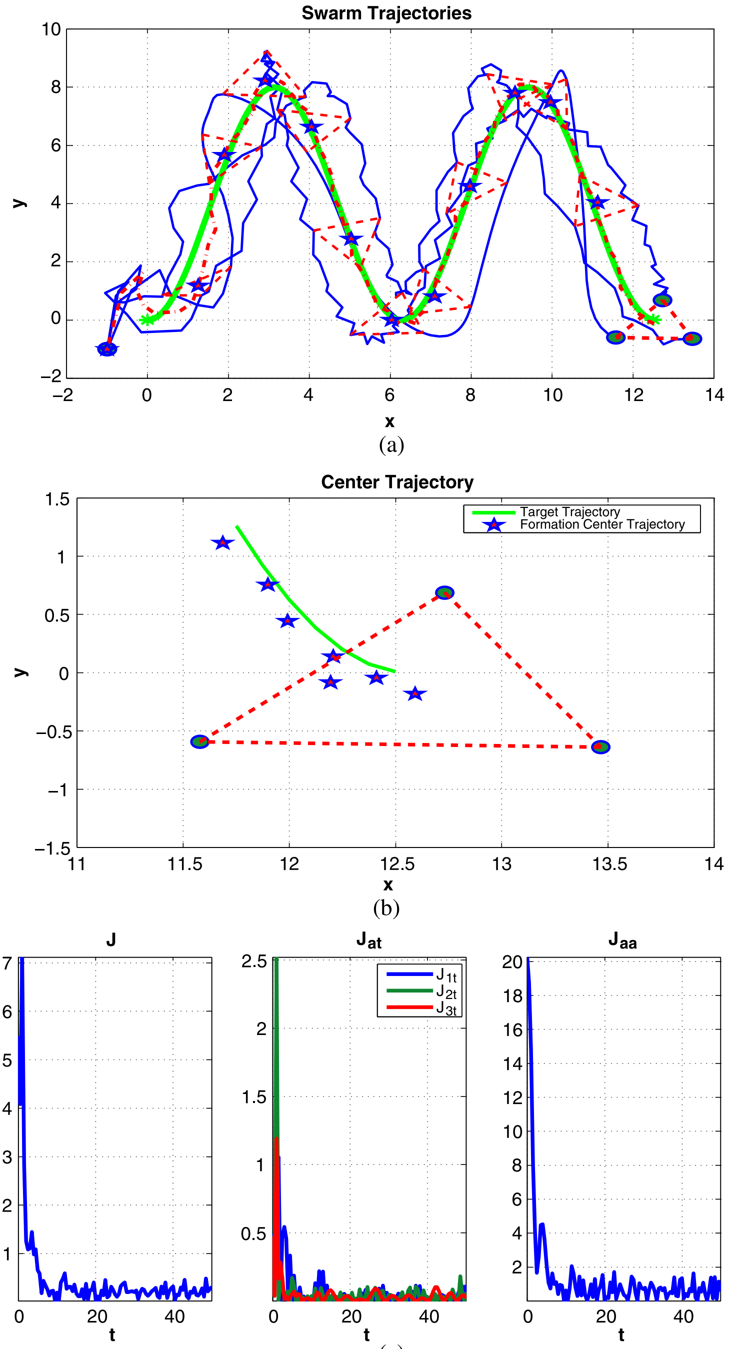


(a)

(b)

(c)

Fig. 8.6 Swarm tracking via perturbation based extremum seeking control, kinematic point-mass, obstacle avoidance: (**a**) trajectories of the swarm agents; (**b**) obstacle avoidance; (**c**) potential functions

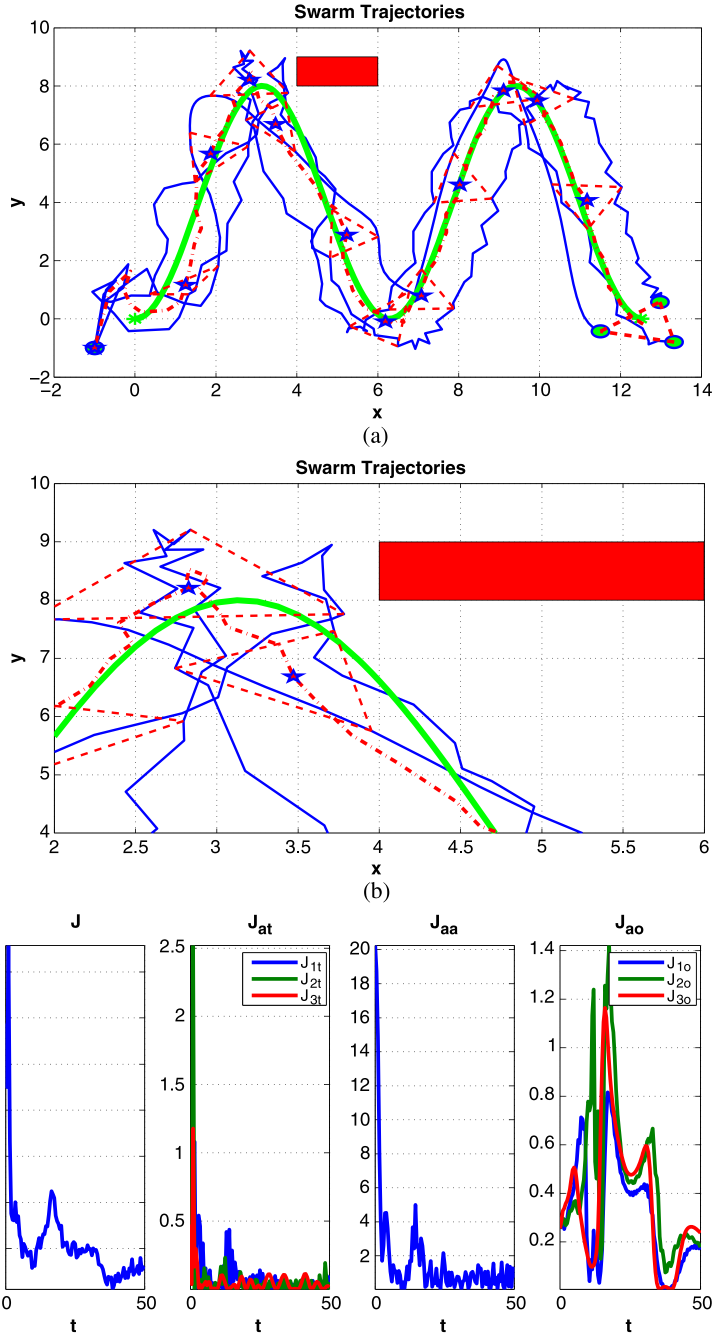


(a)

(b)

(c)

Fig. 8.7 Swarm source seeking via perturbation based extremum seeking control, dynamic point-mass: (**a**) trajectories of the swarm vehicles; (**b**) final formation; (**c**) potential functions



(a)

(b)

(c)

Fig. 8.8 Swarm source seeking via perturbation based extremum seeking control, dynamic point-mass, obstacle avoidance: (**a**) trajectories of the swarm vehicles; (**b**) obstacle avoidance; (**c**) potential functions

and 5, where the set point candidate x^i_{k+1} is generated by the numerical optimization algorithm and the controller for the ith agent can be designed as

$$\begin{aligned} u^i(t) &= -B^\top e^{A^\top (t^i_{k+1}-t)} W_c^{-1}\big(\delta^i_k\big)\big[e^{A\delta^i_k} x\big(t^i_k\big) - x^i_{k+1}\big] \\ &= -\frac{1}{\delta^i_k}\big(x\big(t^i_k\big) - x^i_{k+1}\big), \end{aligned} \tag{8.25}$$

where $t^i_k \le t \le t^i_{k+1}$ and $\delta^i_k = t^i_{k+1} - t^i_k$. The controller (8.25) will regulate the agents to x^i_{k+1} in δ^i_k time, or

$$u^i = -k_i\big(x^i - x^i_{k+1}\big), \quad k_i > 0 \tag{8.26}$$

which will regulate the agents to x^i_{k+1} asymptotically.

8.5.1 *Kinematic Point-Mass Model Simulations: Target Tracking and Formation Orientation*

Here, we use direct search[1] to generate the set point candidate x^i_{k+1} and apply the controller (8.26) for each agent in the simulation, where the setting of the potential functions and initial conditions are the same as in Sect. 8.3. The additional requirement for direct search is to have four performance output measurements in the same time on the corner of the rectangular structure centered at the agent's current position, which can be realized if the agent is equipped with four sensors.

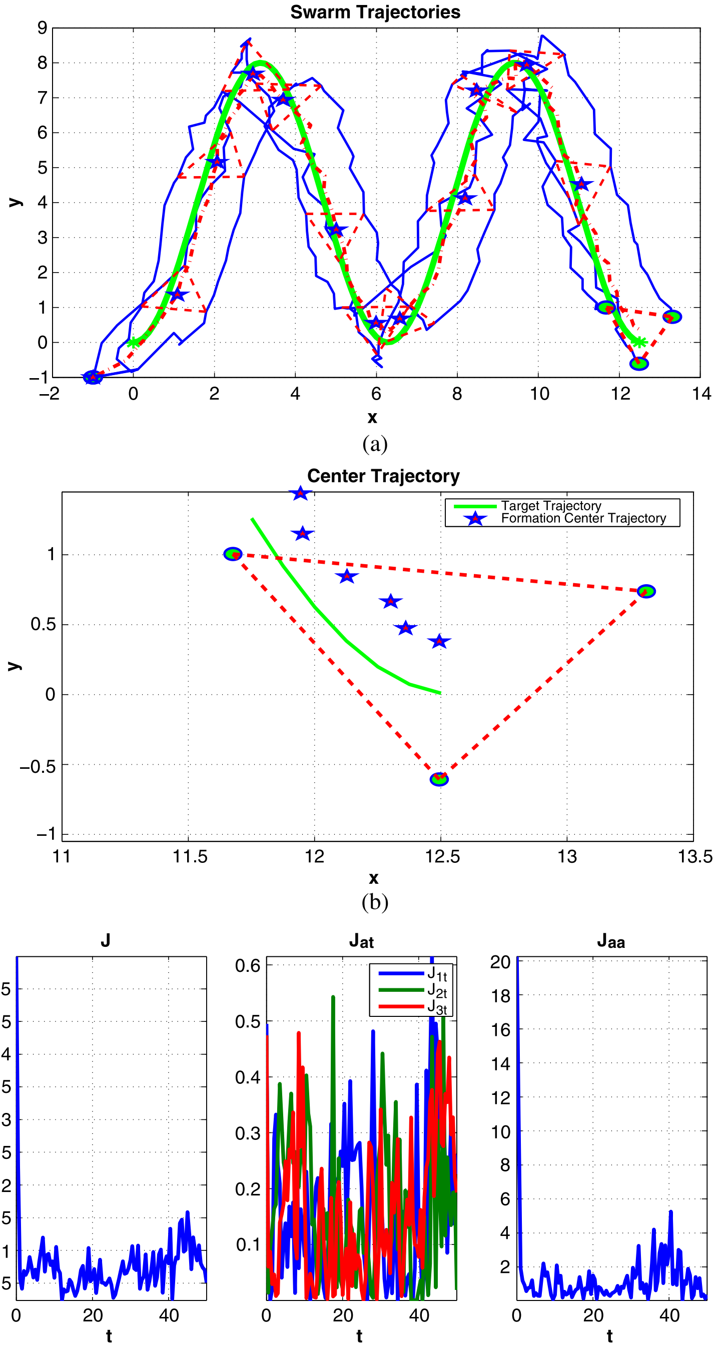
We will consider two cases here: first, similar to Sects. 8.3.2 and 8.4.2, we present simulation results for three-agent swarm tracking using a kinematic point-mass model, and as in Sect. 8.4.3, we also consider tracking in the presence of an obstacle. Then, we augment the potential function with a term that allows us to also control the orientation of the triangular swarm.

8.5.1.1 Swarm Tracking Without Orientation Control

The simulation results can be found in Fig. 8.9, where $\delta_k = 0.005$, $k_i = 5$, $K_{at} = 1$ and $K_{aa} = 0.2$. Other non-gradient numerical optimization based design could be used as well [45].

The simulation results for the obstacle at $o = [4, 6, 8, 9]$ can be found in Fig. 8.10, where $\delta_k = 0.005$, $k = 30$, $K_{at} = 1$, $k_{aa} = 0.2$ and $k_{ao} = 0.05$.

[1] Direct search is used here as a valid optimization method for NOESC because its behavior and properties are similar to those of derivative-free trust region methods. Moreover, since we rely on an asymptotic controller, all stability, convergence and robustness results from Chap. 5 directly apply here.

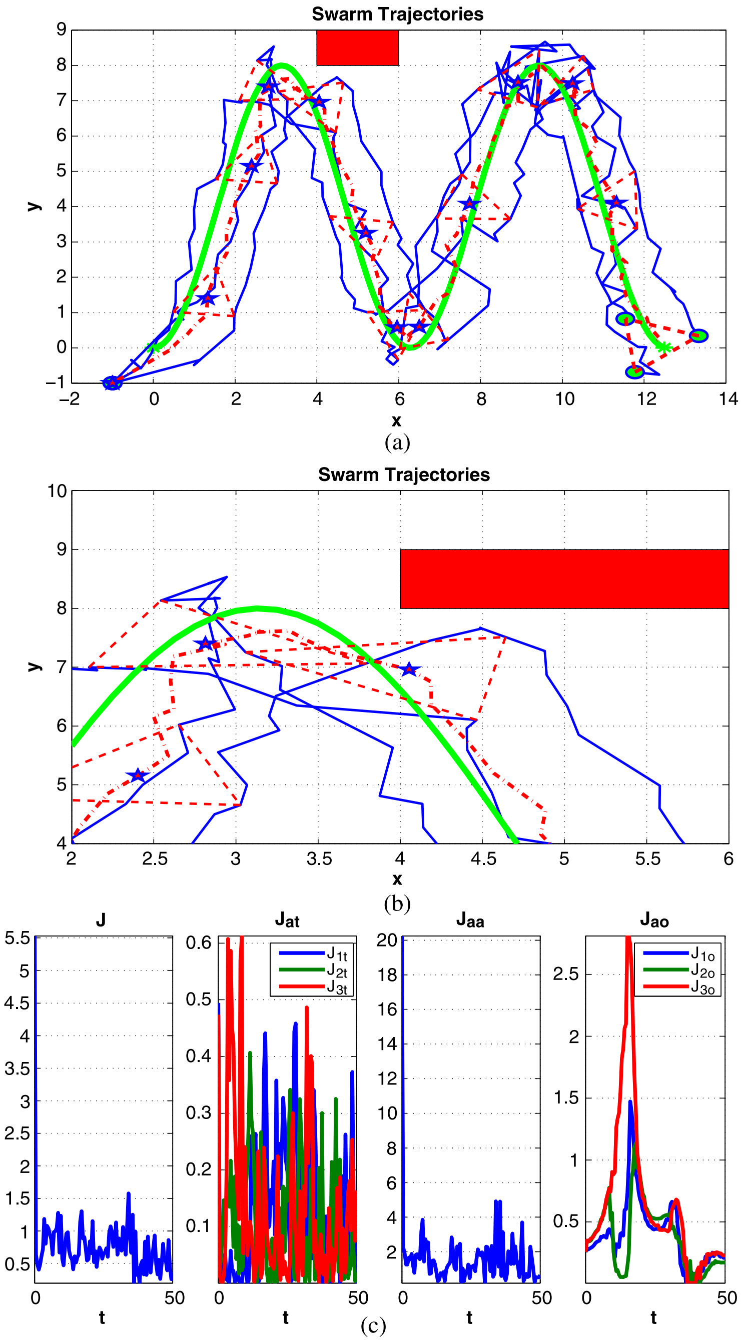


(a)

(b)

(c)

Fig. 8.9 Swarm tracking via numerical optimization based extremum seeking control, kinematic point-mass: (**a**) trajectories of the swarm agents; (**b**) final formation; (**c**) potential functions



(a)

(b)

(c)

Fig. 8.10 Swarm source seeking via numerical optimization based extremum seeking control, kinematic point-mass, obstacle avoidance: (**a**) trajectories of the swarm vehicles; (**b**) obstacle avoidance; (**c**) potential functions

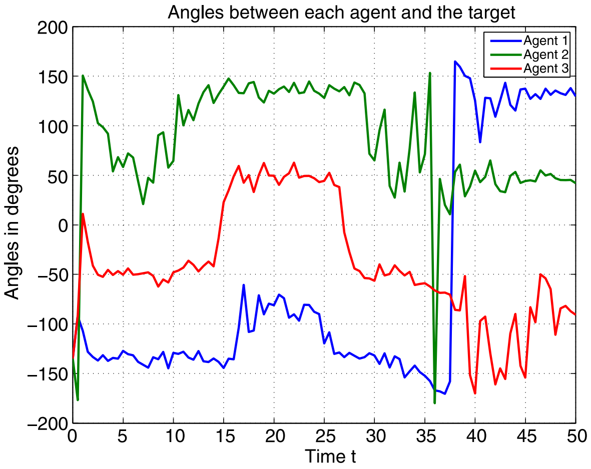


Fig. 8.11 Orientation angles between the agents and the target, without orientation control

8.5.1.2 Swarm Tracking with Orientation Control

Consider the simulation results for swarm tracking of the point-mass kinematic model shown in Fig. 8.9(a). Notice that even though the triangular formation objective is achieved, the formation itself attains a rather unpredictable orientation with respect to a fixed frame of reference. To better illustrate this point, Fig. 8.11 shows a plot of the angles formed between the x-axis and a line joining each agent and the target. As can be clearly seen, the angles are not constant and change wildly as the formation goes through the peaks and valleys of the target's motion.

In practice, one may be interested in being able to not only set the shape of the formation, but also its attitude with respect to the target. For instance, if the agents need to examine the target using an asymmetrical sensor (such as a camera, or a horn-type antenna), then the *orientation* itself of the formation may be critical.

Here, we present a simple method to achieve orientation control, in addition to the other objectives already met (formation control, target tracking and enclosure, and obstacle avoidance), by adding an extra term to the potential function (8.20). In particular, let the potential function be given by

$$\begin{aligned} y &= J(x, x_t) \\ &= K_{at} J_{at}\bigl(x^1, x^2, x^3, x_t\bigr) + K_{aa} J_{aa}\bigl(x^1, x^2, x^3\bigr) + K_{ar} J_{ar}\bigl(x^1, x^2, x^3, x_t\bigr), \end{aligned} \tag{8.27}$$

where K_{ar} is a weight used to prioritize the importance of orientation-keeping, and

$$J_{ar}\bigl(x^1, x^2, x^3, x_t\bigr) = \frac{1}{2} \sum_{i=1}^{3} J_{ir}(x, x_t), \tag{8.28}$$

and one can choose, for example,

$$J_{ir}(x, x_t) = \left[\arctan\left(\frac{y_p^i - x_{t_2}}{x_p^i - x_{t_1}}\right) - \theta_i\right]^2. \tag{8.29}$$

The angles θ_i are *desired* angles between each agent and the target, and are defined with respect to a fixed frame of coordinates. Clearly, these angles need to be chosen to be geometrically consistent with the formation parameters in J_{aa}, as otherwise the controller would be attempting to meet contradicting objectives.

In this simulation, we keep all the parameters chosen in Sect. 8.5.1.1. In addition, we set the orientation potential weight to $K_{ar} = 150$. Note that this number is significantly larger than the other weights; this is due to the different relative magnitudes of radians (small) and distances (large). Finally, we choose the desired orientation angles $\theta_1 = 90°$, $\theta_2 = -30°$ and $\theta_3 = -150°$. As can be observed in Fig. 8.12, the orientation angles are maintained approximately even as target tracking proceeds.

8.5.2 Dynamic Point-Mass Model Simulations

Now, consider the dynamic point-mass model (8.21) in the two-dimensional plane. Let $x^i = [x_p^i, y_p^i, x_v^i, y_v^i]$, where x_p^i, y_p^i denotes the position and x_v^i, y_v^i denotes the velocity of the ith agent, respectively. Note that this time the performance is a function of the position only, thus we need to design the control $u^i = [u_x^i, u_y^i]$ to regulate the position state $[x_p^i, y_p^i]$ to the set point $[x_{k+1,x}^i, x_{k+1,y}^i]$.

Let

$$e_x^i = k_e^i\big(x_p^i - x_{k+1,x}^i\big) + x_v^i$$

and

$$e_y^i = k_e^i\big(y_p^i - x_{k+1,y}^i\big) + y_v^i.$$

Then,

$$\dot{e}_x^i = k_e^i v_x^i + u_x^i$$

and

$$\dot{e}_y^i = k_e^i v_y^i + u_y^i.$$

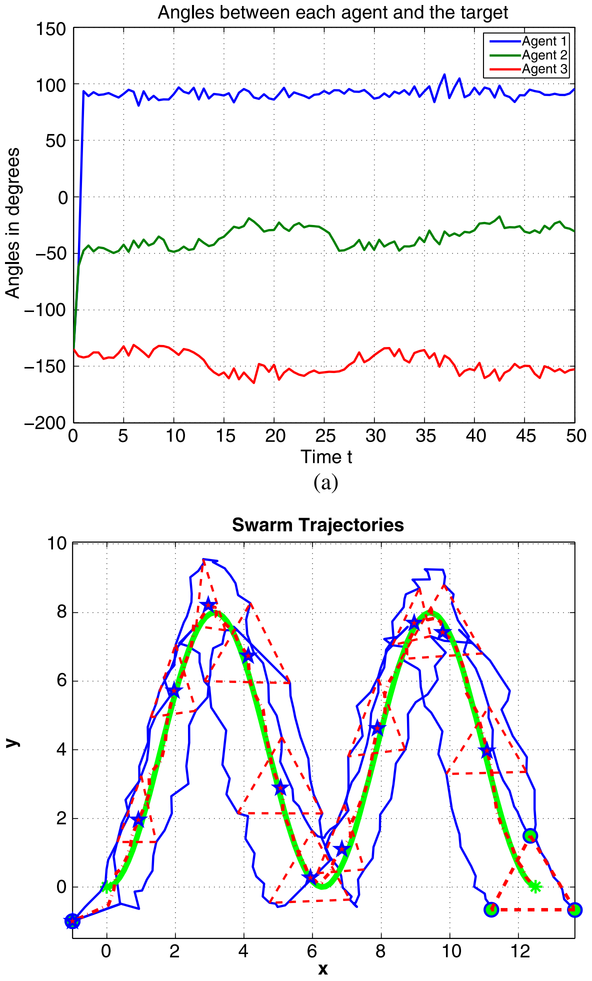
We can design the controller for the ith agent modeled by a double integrator as

$$u^i = \big[u_x^i, u_y^i\big]^\top = \big[-k_e^i v_x^i - k_i e_x^i, -k_e^i v_y^i - k_i e_y^i\big]^\top, \tag{8.30}$$

where $k_e^i, k_i > 0$. The simulation results for a dynamic point mass model can be found in Fig. 8.13, where $\delta_k = 0.01$, $K_{at} = 1$, $K_{aa} = 0.2$, $k_1 = 10$ and $k = 20$.

Moreover, the controller design (8.30) is robust to a bounded input disturbance due to the damping term $-k^i e_x^i$ and $-k^i e_y^i$. Simulation results of the obstacle avoid-

Fig. 8.12 Swarm source seeking via numerical optimization based extremum seeking control, kinematic point-mass, orientation control: (**a**) orientation angles between the agents and the target; (**b**) trajectories of the swarm vehicles



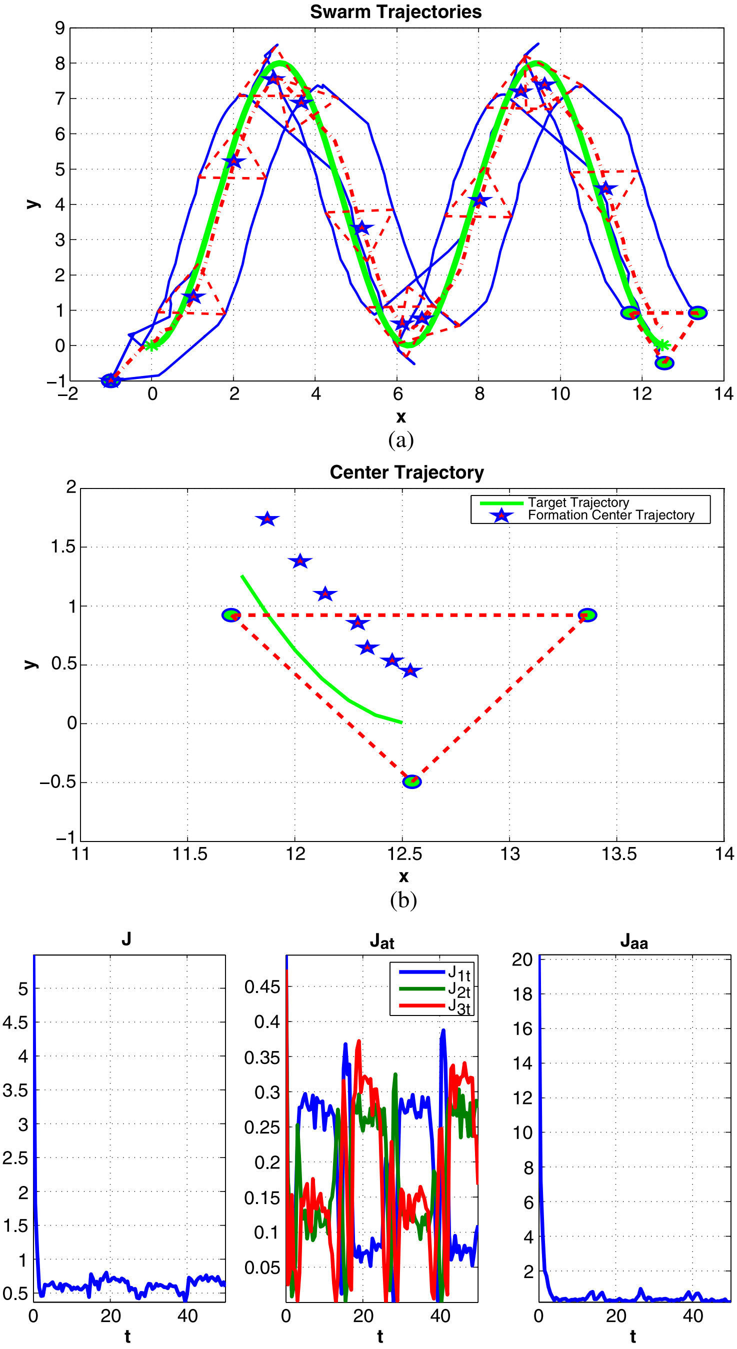
(a)

(b)

ance case with unknown input disturbance $5\sin(0.2t)$ can be found in Fig. 8.14, where $\delta_k = 0.005$, $K_{at} = 1$, $K_{aa} = 0.3$, $K_{ao} = 0.05$, $k_e^i = 10$ and $k_i = 10$ for all $i = 1, 2, 3$. Given the weights selection, this design focuses more on collision avoidance and formation control. At the same time, we still enclose the moving target regardless of the input disturbance and more complicated dynamical model.

8.6 Application: Detection of Leakage Points Using a Mobile Radar Sensor Network

In this chapter we have studied a variety of formation control methods using extremum seeking control. A possible application of these methods is the localization

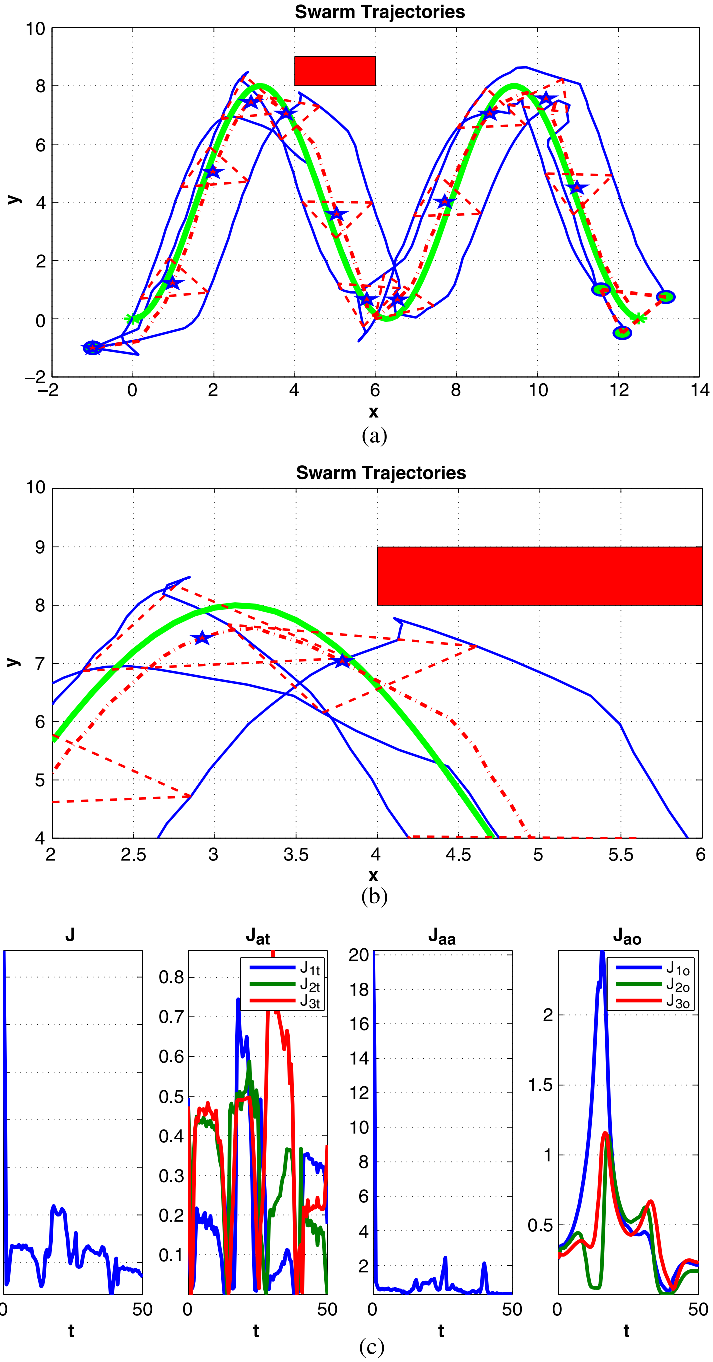


(a)

(b)

(c)

Fig. 8.13 Swarm source seeking via numerical optimization based extremum seeking control, dynamic point-mass: (**a**) trajectories of he swarm vehicles; (**b**) final formation; (**c**) potential functions



(a)

(b)

(c)

Fig. 8.14 Swarm tracking via numerical optimization based extremum seeking control, dynamic point-mass, obstacle avoidance with input disturbance: (**a**) trajectories of the swarm agents; (**b**) obstacle avoidance; (**c**) potential functions

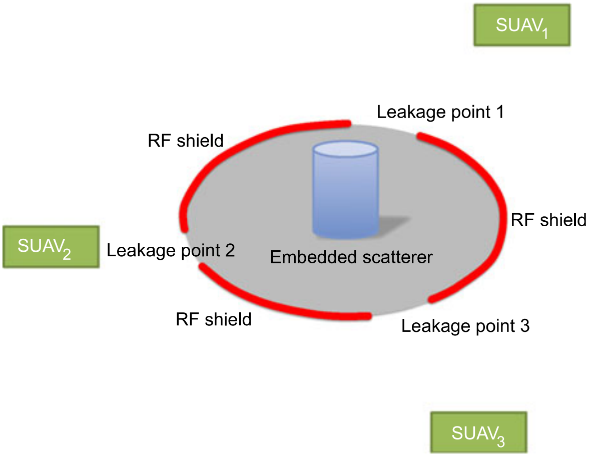


Fig. 8.15 Embedded scatterer surrounded by RF shields, with leakage points

of RF "leakage points" using a swarm of small unmanned aerial vehicles (SUAVs). The work this material is based on originally appeared in [44], and related concepts have been explored in [2, 15, 31, 32, 37].

8.6.1 Problem Statement

The main idea is illustrated in Fig. 8.15, where an "embedded scatterer" can be seen in the center, surrounded by three metal shields, with three "holes," or leakage points, between them. The nature of the scatterer is not known, and the objective is to locate the leakage points using a collaboration between a group of SUAVs. Each SUAV is equipped with an active radar sensor, whose measurements are used to help guide the SUAV to the vicinity of a signal leakage point.

The number of leakage points and their locations are not known in advance, and thus it is conceivable that in some conditions, some leakage points may not be found, while others may be found by more than one SUAV. In order to maximize the chances of finding all points and minimizing the likelihood of more than two vehicles finding the same point, we will use the general potential function approach described in Sect. 8.2, where the radar measurements obtained by each SUAV will be directly used as part of the potential function.

Because the part of the potential function derived from the radar measurements is completely unknown, and highly noisy, it becomes unfeasible to use a GESC approach, as in Sect. 8.3, since it is impossible to construct an analytical expression of the gradient. In the following, we investigate the use of NOESC to solve the problem of leakage point location, by extending the simulations from Sect. 8.5. Since the potential function is not expected to be smooth, we will use direct search as the numerical optimizer.

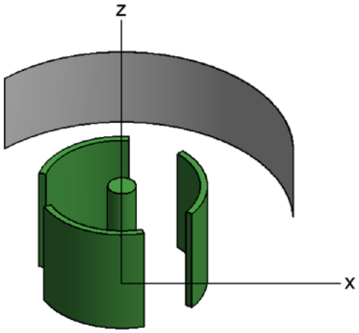


Fig. 8.16 Geometry used for electromagnetic simulations

8.6.2 RF Signal Synthesis for Multi-SUAV Geometry

In order to proceed with this application, radar signals need to be synthesized for a notional set of SUAV geometries via electromagnetic simulation.

Figure 8.16 illustrates the geometry used to conduct electromagnetic simulations of the multi-SUAV example depicted in Fig. 8.15. Electromagnetic simulations for purposes of this investigation were conducted using the software package FEKO [22]. FEKO offers several options for performing electromagnetic simulations, including a method-of-moments code (MoM), a physical optics code (PO), and hybridized versions of these two codes. We selected the PO code due to time constraints and the need for computational efficiency. In order to simulate scattering from the perfect electric conductors (PEC) within the geometry of Fig. 8.16, consider the surface scattering integral

$$E^s = \frac{k\eta_0}{4}\int\!\!\int J_s\left(\vec{r}'\right) g\left(\vec{r},\vec{r}'\right) dS'. \tag{8.31}$$

In (8.31), J_s is the surface current density on the PEC materials within our simulated environment, $\vec{r}'$ represents localized coordinates on the surface of the PEC scatterers, $\vec{r}$ represents field evaluation coordinates (i.e. the synthesized received waveform), η_0 is the free-space impedance, k is the free-space wavenumber, and g is the free-space Green's function given by

$$g(\vec{r},\vec{r}') = \frac{\exp(jkR)}{4\pi R},$$

where R is the distance. The PO or Kirchhoff approximation for (8.31) is

$$J_s = \hat{n} \times \left(H^i + H^s\right) \approx 2\hat{n} \times H^i. \tag{8.32}$$

Equation (8.32) can greatly simplify the numerical computations (i.e., it allows us to justify using a PO code) and implies that the surface magnetic field, H^s, is approximately equal to the incident magnetic field, H^i from the SUAV source. Meeting this surface boundary condition requirement motivates the selection of relatively large (or smooth) structures in relation to the wavelengths for the radar waveforms that we select for our multi-platform SUAV simulation.

In order to continue generating the parameters and pre-conditions for this simulation set, we selected a design simulation frequency for each of the three notional radars on the three SUAV platforms of 2 GHz. This frequency corresponds to a wavelength of 15 cm at the selected center frequency of 2 GHz. We also selected a bandwidth of 100 MHz to simulate each of the three SUAV radars. For our case, this corresponds to 100 frequency points from 1.95 to 2.05 GHz. Next, we select the PEC structures for our scattering environment to be relatively large in relation to our characteristic wavelength of 15 cm. Considering this constraint/approximation, we selected dimensions for the cylindrical scatterer at the center of the simulation space in Fig. 8.16 as 2 meters in height and 1 meter in diameter. The three shields, used to define the leakage points, each have a height of three meters. The "holes" or leakage points between the three shields are 25.5 degrees each, and the radius from the center of the cylinder to the boundary of any given shield is 3 meters.

Considering these structural dimensions, along with the physical constraints for accurate PO simulation, we selected a simulation grid size of one-third wavelength, or 5 cm. This selection was partially justified by the fact that all the structures in our simulation space are smooth, and all the structures along with all the other gaps between structures are large compared to 5 cm. In order to verify the validity of this approach, we produced several sample simulations at one-sixth the wavelength for a grad spacing and compared outputs. The resulting outputs from two initial sample runs were identical to within over a 95 percent accuracy level.

After selecting the above-mentioned simulation parameters, we conducted a set of simulations for the PEC based scattering environment of Fig. 8.16 with the FEKO PO code as a function of azimuth angle and elevation angle. We simulated scattering output corresponding to 100 frequency points, from 1.95 to 2.05 GHz for a series of spatial points within the simulation space, where we varied the elevation angle from 30 degrees to 3 degrees at three degree increments, and the azimuth angle from 0 to 120 degrees at 1 degree increments. We generated 360 degrees of simulation data in azimuth due to the symmetries within our simulation space. After generating this set of synthetic data, we ported these data into a MATLAB workspace and conditioned the data to simulate scattering versus range and azimuth, by applying the appropriate link budget based terms for propagation loss due to $1/R^4$ terms from the basic radar range equation. Thus, via this conversion from spherical to cylindrical coordinates, effectively we can assume to have data in terms of azimuth (denoted by ϕ) and range (denoted by R).

All SUAV platform altitudes are selected at 3 meters (across all azimuth and elevation simulation points) for these signal conditioning computations. While the resulting signal simulation is relatively coarsely spaced in terms of the spatial increments between neighboring points, we have a large enough density of points to meet the basic objectives of our investigation.

8.6.3 NOESC for Leakage Point Localization

The key element that allows us to use the general swarming theory described in this chapter to solve this particular problem is a suitable choice of potential function. In what follows, we describe the potential function by decomposing it into its motion and avoidance components (which are similar to those used in the examples we have already seen), and a new component specific to this problem that is based on the radar measurements performed by each SUAV.

More specifically, let the potential function for the ith vehicle be given by

$$J_i(x, x_t) = K_s J_{is}\big(x^i, x_t\big) + K_{aa} J_{ia}(x) + K_r J_{ir}\big(x^i\big), \tag{8.33}$$

where J_{is} is a motion-control component, J_{ia} is an inter-vehicle avoidance component, and J_{ir} is a term computed directly from the radar measurements. The corresponding weights K_s, K_{aa} and K_r allow the designer to balance the behaviors encoded by each term in (8.33) (we choose them equal for all vehicles for simplicity). Note that here each vehicle has its own potential function, independent of the other vehicles—thus, this control scheme can easily be implemented in a decentralized fashion. Of course the inter-vehicle distances are required to compute $J_{ia}(x)$, but in practice these can be obtained by each individual vehicle via the use of a range finder, if inter-vehicle communication is not desired.

8.6.3.1 Motion and Avoidance Potentials

In this application, we make the simplifying assumption that the location of the central cylindrical scatterer in Fig. 8.16 is at least approximately known. Also, we assume some initial estimates for the distance between the metal shields and the cylinder. With these assumptions, we are able to direct the SUAVs to move in the general direction of the embedded scatterer, but without approaching it so much that a collision could take place.

With these assumptions in mind, let the motion-control component of the potential function for the ith vehicle be given by

$$J_{is}\big(x^i, x_t\big) = \big[\big(x_p^i - x_{t_1}\big)^2 + \big(y_p^i - x_{t_2}\big)^2 - \delta_t^2\big]^2, \tag{8.34}$$

where x_t is the known location of the central cylinder. The potential (8.34) compels the ith SUAV to move toward the target x_t while getting no closer than a distance δ_t (chosen to be the same for all SUAVs, for simplicity).

As the SUAVs explore the area around the target looking for radar leakage points, it is imperative to prevent them from colliding with each other. Thus, we define the inter-vehicle collision avoidance potential for the ith vehicle as

$$J_{ia}(x) = \sum_{j=1, j\neq i}^{N} J_{ij}\big(\big\|x^i - x^j\big\|\big), \tag{8.35}$$

where

$$J_{ij}\big(\|x^i - x^j\|\big) = \frac{1}{\|x^i - x^j\|^4}. \tag{8.36}$$

The reason for the choice (8.36) for inter-vehicle collision avoidance is that this function has very small magnitude for large distances—in practice, it has a negligible effect as long as the vehicles are in no danger of colliding. But the magnitude of (8.36) increases rapidly as the distance decreases, and it will quickly become larger than the other terms, a necessary condition if collision is to be avoided. The potential (8.36) is equal to 1 when $\|x^i - x^j\| = 1$ meter, thus defining a radius of action. This radius can be adjusted as desired via a suitable weight of the denominator of (8.36).

8.6.3.2 Radar Measurement Potential

The most important part of the potential function formulation for this problem is the term related to radar measurements. For the ith SUAV, let the function $F_{ir}(f, \phi, R)$ denote the measurements obtained from performing a frequency scan (from 1.95 to 2.05 GHz) with a single radar antenna, where f is the frequency, and the SUAV the antenna is mounted on is located at azimuth ϕ and range R with respect to the embedded scatterer. Let $\mathcal{F}^{-1}$ denote the inverse Fourier transform. Then, let

$$f_{ir}(t, \phi, R) = \mathcal{F}^{-1}\big\{F_{ir}(f, \phi, R)\big\},$$

and compute

$$\bar{f}_{ir}(\phi, R) = \max_t f_{ir}(t, \phi, R). \tag{8.37}$$

Equation (8.37) converts the raw synthetic data from the frequency, azimuth, and range domain into the time domain via inverse Fourier-transforming each point-by-point frequency slice into a synthesized temporal domain and then extracting the maximum value from each temporal slice of data. This operation transforms the raw data from a basic three-dimensional simulation space into a two-dimensional array of maximum values as a function of azimuth and range.

Next, let $H_\phi(\cdot, w)$ denote a w-element sliding low-pass filtering operation across azimuth, and

$$\bar{f}_{ir_{LP}}(\phi, R) = H_\phi\big(\bar{f}_{ir}(\phi, R), w\big). \tag{8.38}$$

Equation (8.38) has the effect of generating a smoothed RF data array with relatively larger azimuth-dependent scattering trends from the shield structures and relatively smaller azimuth-dependent scattering trends from within the leakage points.

Note that (8.38) assumes the ability to perform a filtering operation over a w-window of azimuth angles (at 1 degree increments, due to the simulation conditions described in Sect. 8.6.2). In practice, this means that each SUAV would need to perform w frequency scans at w contiguous azimuth angles. This is not an unreasonable assumption, as all it implies is a delay proportional to the size w of the window in order to be able to compute the value of the potential function.

The final step in the computation of the potential again makes use of the sliding w-window, where we let

$$J_{ir}(\phi, R) = \Big[\max_{\phi_{-w} \leq \phi \leq \phi_w} \big(\bar{f}_{ir_{LP}}(\phi, R) \big) - \min_{\phi_{-w} \leq \phi \leq \phi_w} \big(\bar{f}_{ir_{LP}}(\phi, R) \big) \Big]^2 \tag{8.39}$$

denote the value of the radar-related potential function, evaluated at cylindrical coordinates (ϕ, R) with respect to the center of the embedded scatterer. The max and min operations are performed on the azimuth interval $\phi_{-w} \leq \phi \leq \phi_w$, where ϕ_{-w} and ϕ_w are the lower and upper limit, respectively, of the w-sized sliding window centered on ϕ. Clearly, (8.39) can be evaluated at any rectangular coordinate $x^i = [x_p^i, y_p^i]^\top$ of the SUAV via a suitable coordinate transformation. Thus, using a slight abuse of notation, we will write (8.39) interchangeably as $J_{ir}(\phi, R)$ or $J_{ir}(x^i)$.

8.6.4 Results

In order to better visualize what the potential (8.39) does and the effect of the sliding window size, consider Fig. 8.17, where (8.39) is plotted for ten different values of range R and $0° \leq \phi < 360°$, with different choices for window size w. As one could expect, the larger w is, the more clearly the leakage points can be detected, with fewer local minima in J_{ir}.

In the simulations below, we present results for $w = 10$ and $w = 4$, for comparison purposes. We use the kinematic model (8.1) to represent the SUAVs (this is clearly a great simplification with respect to a real application, but we do it so that we can focus on the potential function choice), together with controller (8.26). The simulation parameters for NOESC are $K_s = 1$, $K_{aa} = 5 \times 10^8$, $K_r = 0.001$, $k_i = 0.1$, $\delta_k = 10$ seconds, and we perform all simulations for 1500 seconds, with $N = 3$ (that is, we use three vehicles).

Figure 8.18 shows the first simulation, where the window size is $w = 10$. In Fig. 8.18(a) one can see the paths taken by the vehicles, whose initial conditions are marked next to each vehicle's label. The scattering cylinder is located on coordinate $[15, 25]^\top$, and the surrounding metal shields are also shown in red. Finally, a scaled version of (8.39) is shown for all vehicles, for illustration purposes. The color of the plot of (8.39) matches the color of the trace of each vehicle.

Figure 8.18(b) shows plots of the individual potential functions for each vehicle, weighted by their respective constants in order to compare the relative scales. Note that the inter-vehicle collision avoidance terms increase, as expected, as the vehicles start to approach each other. Figure 8.18(c) shows an aggregation of the potentials, where $J_s = \sum_i J_{is}$, $J_{aa} = \sum_i J_{ia}$ and $J_r = \sum_i J_{ir}$. The aggregates are again weighted by their corresponding constants, thus facilitating a comparison of the relative effect of each term as the vehicles approach the target.

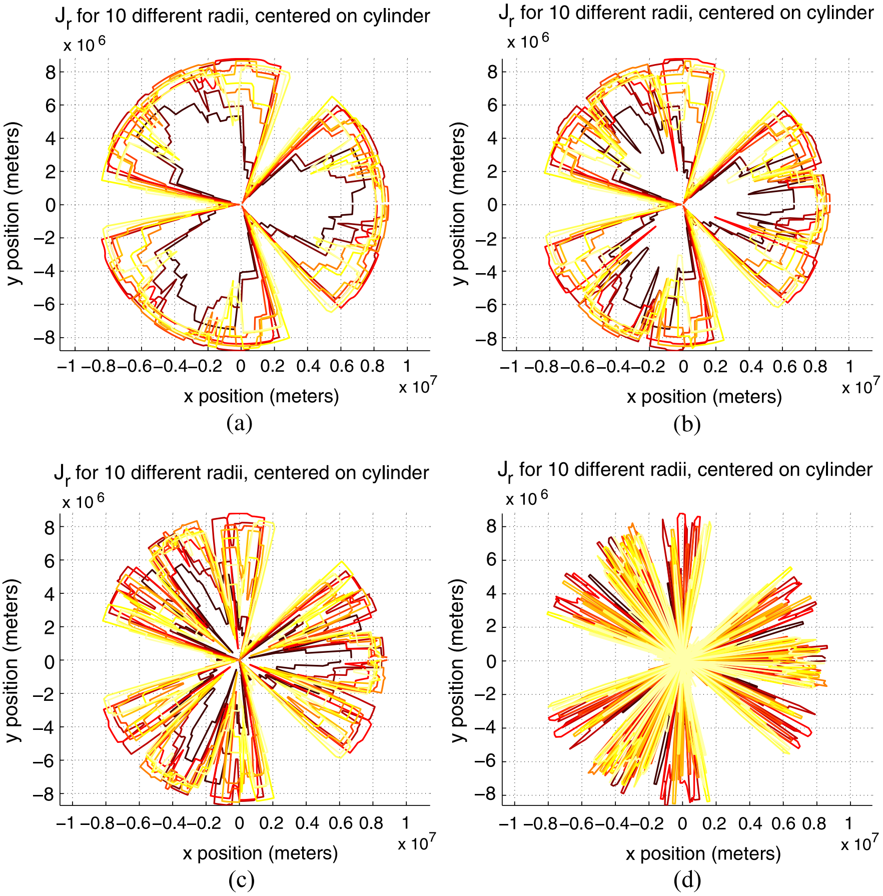
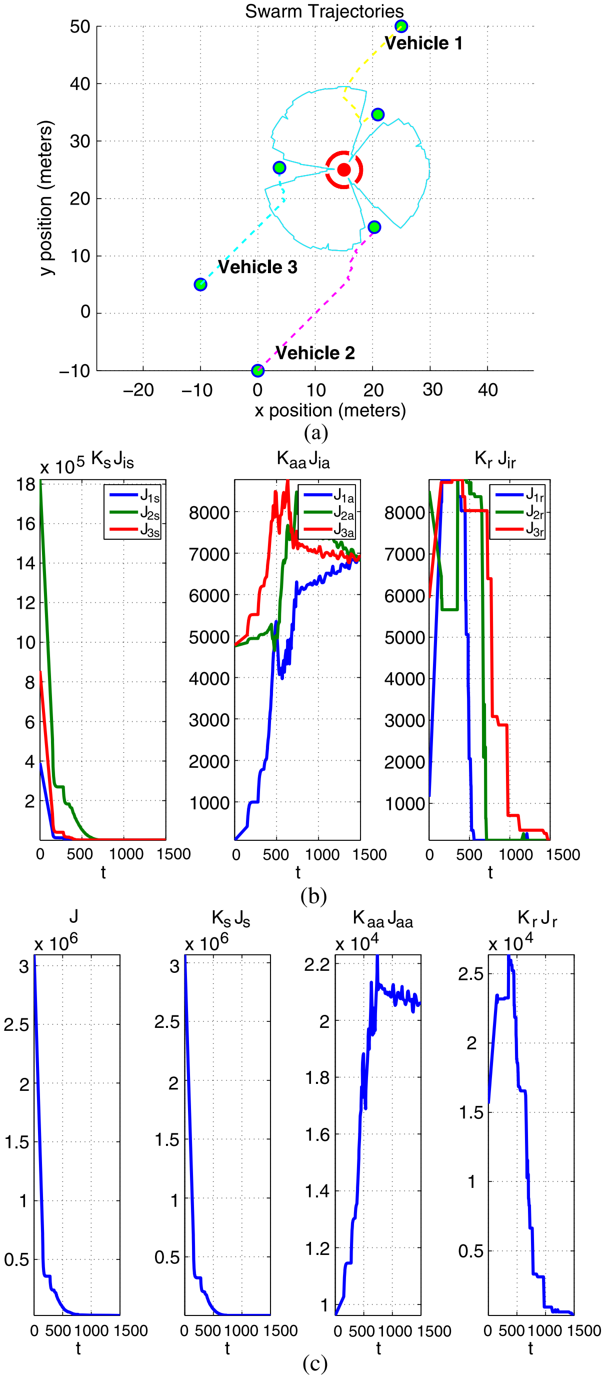


Fig. 8.17 Radar potential (8.39) for ten different radii and varying window size: (**a**) $w = 10$; (**b**) $w = 7$; (**c**) $w = 4$; (**d**) $w = 1$

For the initial vehicle positions chosen in the simulation of Fig. 8.18 we see that the three leakage points are found. The same is true in Fig. 8.19, where now $w = 4$ but the same vehicle initial positions are chosen.

Figures 8.20 and 8.21 present a different scenario, where $w = 10$ in the first case and $w = 4$ in the second. Here, when the larger $w = 10$ window is used, the three leakage points are again found; but when $w = 4$, we observe vehicles 1 and 3 getting stuck in local minima of J_{1r} and J_{3r}, respectively.

The last scenario is shown in Fig. 8.22, where only the $w = 10$ case is presented. For this initial vehicle positions we see that only two leakage points are found. All vehicles are simply too far from the minimum of the third leakage point, and are unable to locate it. A case like this indicates that, in a practical application of this scheme, the designer may want to augment the controller with some logic that, for instance, makes decisions to send a vehicle to an entirely different area based on relative values of J_{ir} when steady-state has been reached.

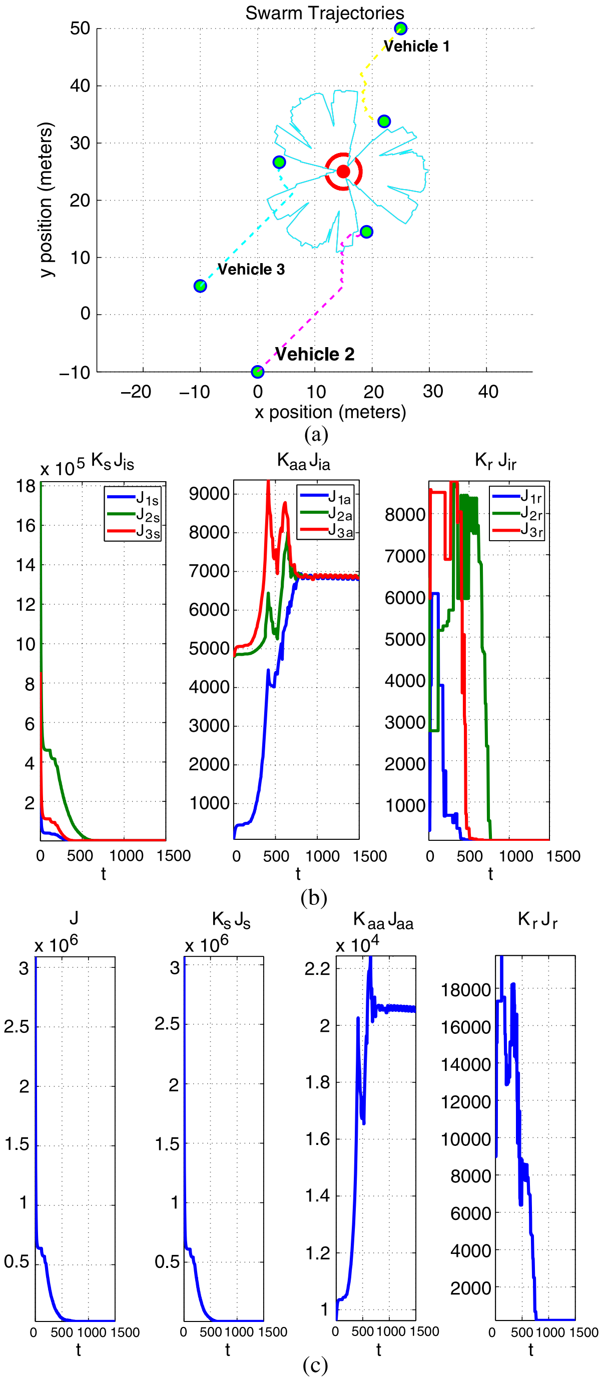


(a)

(b)

(c)

Fig. 8.18 Leakage point localization using NOESC with $w = 10$: (**a**) vehicle trajectories; (**b**) each vehicle's potential functions; (**c**) aggregated potentials



(a)

(b)

(c)

Fig. 8.19 Leakage point localization using NOESC with $w = 4$: (**a**) vehicle trajectories; (**b**) each vehicle's potential functions; (**c**) aggregated potentials

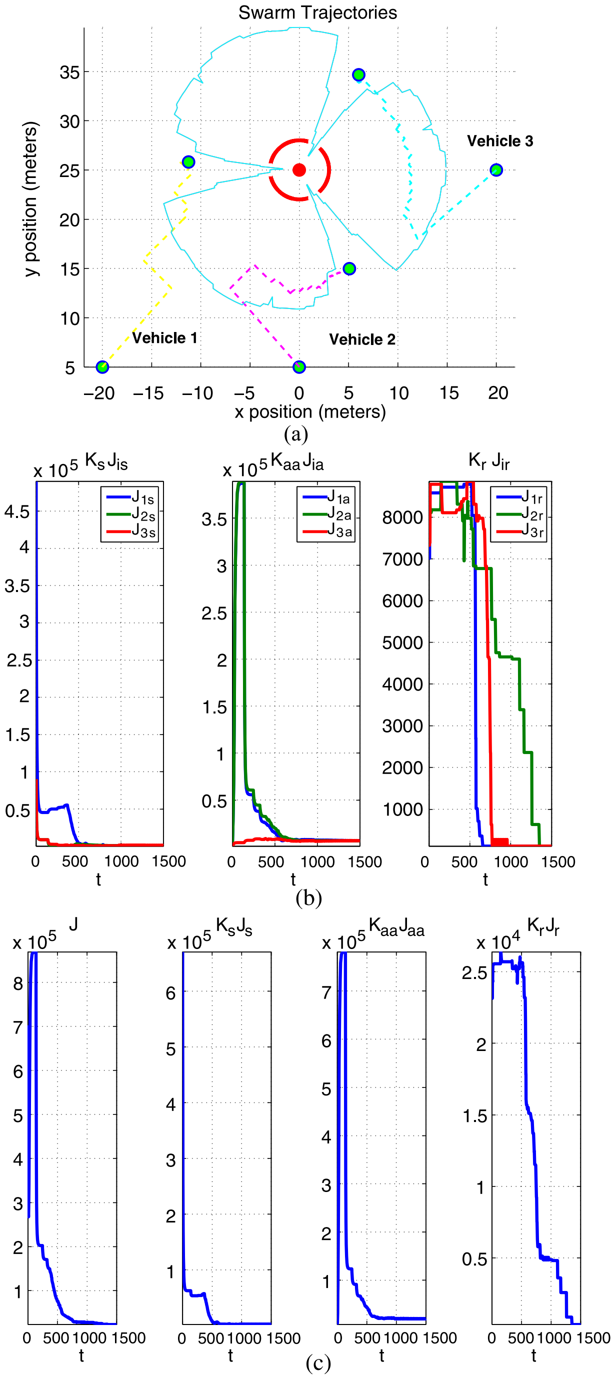
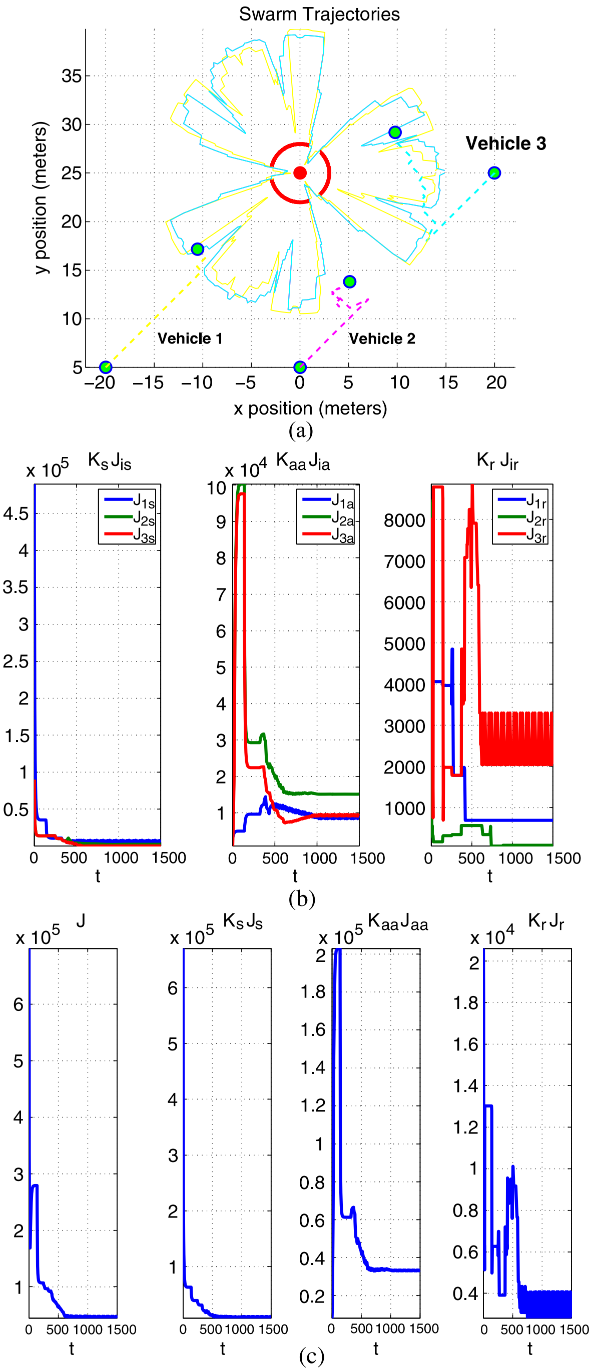
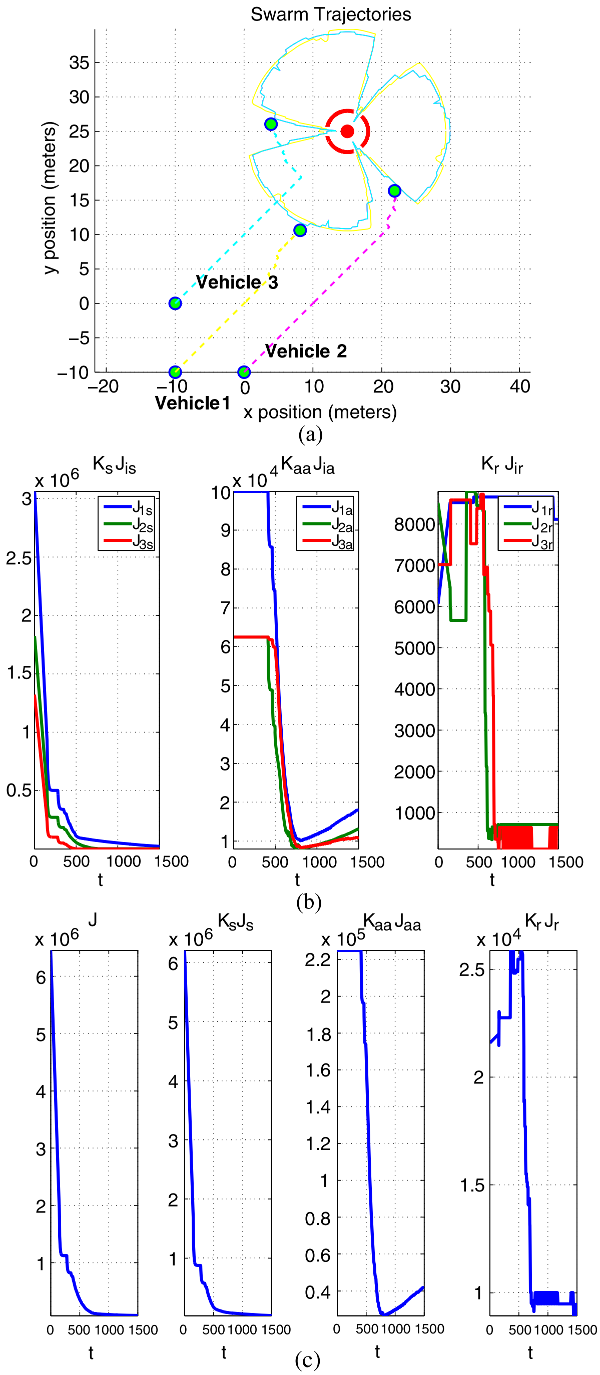


Fig. 8.20 Leakage point localization using NOESC with $w = 10$: (**a**) vehicle trajectories; (**b**) each vehicle's potential functions; (**c**) aggregated potentials

Fig. 8.21 Leakage point localization using NOESC with $w = 4$: (**a**) vehicle trajectories; (**b**) each vehicle's potential functions; (**c**) aggregated potentials





(a)

(b)

(c)

Fig. 8.22 Leakage point localization using NOESC with $w = 10$: (**a**) vehicle trajectories; (**b**) each vehicle's potential functions; (**c**) aggregated potentials

8.7 Concluding Remarks

In this chapter, we present three swarm tracking designs for a group of agents via artificial potential and extremum seeking control. All three extremum seeking control methods achieve source seeking, formation control and collision avoidance, and all the designs can be decentralized if each agent controller is based on its own performance function. The gradient based extremum seeking control design obtains the best performance based on its requirement of gradient information, which is equivalent to knowing the target position. It can be used to avoid obstacles as long as we can calculate the gradient field of the potential between the obstacle and the agents. However, it is generally impossible for irregular and unknown obstacles. The perturbation based and numerical optimization based designs both achieve source seeking, formation control, and collision avoidance with good performance. The obstacle avoidance is also easy to realize via the incorporation of additional potential measurements between the obstacle and the agents due to the non-gradient extremum seeking ability. And such designing allows for unknown and irregular obstacle as long as one can obtain the potential value via sensor measurements.

The non-gradient numerical optimization based extremum seeking design may require more sensor information as we use direct search in this chapter (trust region methods could also be used), and the results are local since the numerical optimization algorithm only provides local convergence. Using recent advances in perturbation based extremum seeking design, one may be able to achieve better tracking performance by jumping over certain local minima by carefully design the amplitude of perturbation signal [41]. The extension of perturbation based extremum seeking control design to more complicated vehicle dynamics is not straightforward as shown in [7–11, 20, 48, 49], and additional robust design issues are not systematically addressed. The numerical optimization based design is relatively straightforward to fit for a more general model of autonomous vehicle, and the robust and adaptive design techniques for numerical optimization based extremum seeking provide large space to accommodate input disturbance, and unmodeled plant dynamics [47].

By observing the simulation results, we found that the control gain of the perturbation based design tends to be high due to the tracking of the moving target. Moreover, something similar occurs for the numerical optimization based design when one desires to improve tracking speed by reducing the regulation time, which also means an increase of the control gain. Thus, we need to have an estimation of the velocity of the target to choose a suitable regulation time δ_k. In the obstacle avoidance case, unlike the continuous optimization running in the gradient based or perturbation based design, the numerical optimization based design does not guarantee that the agent will avoid the obstacle during the transient due to the fact that no potential force is fed back in the regulation period.

Finally, Sect. 8.6 presents a specific application of the general swarming theory described in this chapter. Here, we show how a mobile sensor network can be programmed to localize radar leakage points in a radar scattering environment. The key to achieving this result is in the design of an appropriate potential function that is directly derived from the radar sensor measurements.

References

1. Ariyur, K.B., Krstić, M.: Real-time Optimization by Extremum-seeking Control. Wiley-Interscience, Hoboken (2003)
2. Barber, C., Gates, M., Selmic, R., Issa, H.A., Ordóñez, R., Mitra, A.: PADF RF localization experiments with multi-agent caged-MAV platforms. In: SPIE DSS (2011)
3. Biyik, E., Arcak, M.: Gradient climbing in formation via extremum seeking and passivity-based coordination rules. In: Proceedings of the Conference on Decision and Control, pp. 3133–3138 (2007)
4. Cochran, J., Kanso, E., Krstic, M.: Source seeking for a three-link model of fish locomotion. In: Proceedings of the American Control Conference, pp. 1808–1813 (2009)
5. Cochran, J., Krstić, M.: Source seeking with nonholonomic unicycle without position measurements and with tuning of angular velocity—part i: Stability analysis. In: Proceedings of the Conference on Decision and Control, pp. 6009–6016 (2007)
6. Cochran, J., Krstić, M.: Source seeking with nonholonomic unicycle without position measurements and with tuning of angular velocity—part ii: Applications. In: Proceedings of the Conference on Decision and Control, pp. 1951–1956 (2007)
7. Cochran, J., Krstić, M.: Extremum seeking for motion optimization: From bacteria to nonholonomic vehicles. In: Chinese Control and Decision Conference (2008)
8. Cochran, J., Krstić, M.: 3-d source seeking for underactuated vehicles without position measurement. IEEE Trans. Robot. **25**(1), 117–129 (2009)
9. Cochran, J., Krstić, M.: Nonholonomic source seeking with tuning of angular velocity. IEEE Trans. Autom. Control **54**(4), 713–731 (2009)
10. Cochran, J., Ghods, N., Siranosian, A., Krstić, M.: 3d source seeking for underactuated vehicles without position measurement. IEEE Trans. Robot. **25**, 117–129 (2009)
11. Cochran, J., Kanso, E., Kelly, S.D., Xiong, H., Krstić, M.: Source seeking for two nonholonomic models of fish locomotion. IEEE Trans. Robot. **25**(5), 1166–1176 (2009)
12. Desai, J.P., Ostrowski, J., Kumar, V.: Controlling formations of multiple mobile robots. In: Proc. 1998 IEEE Int. Conf. Robotics and Automation, Leuven, Belgium, pp. 2864–2869, May 1998
13. Fu, L., Özgüner, Ü.: Extremum-seeking control in constrained source tracing with nonholonomic vehicles. IEEE Trans. Ind. Electron. **56**(9), 3602–3608 (2009)
14. Fu, L., Özgüner, Ü.: Sliding mode in constrained source tracking with non-holonomic vehicles. In: International Workshop on Variable Structure Systems (VSS'08), pp. 30–34 (2008)
15. Gates, M., Barber, C., Selmic, R., Issa, H.A., Ordóñez, R., Mitra, A.: PADF RF localization criteria for multi-model scattering environments. In: SPIE DSS (2011)
16. Gazi, V., Ordóñez, R.: Target tracking using artificial potentials and sliding mode control. Int. J. Control **80**(10), 1626–1635 (2007)
17. Gazi, V., Passino, K.M.: A class of attraction/repulsion functions for stable swarm aggregations. Int. J. Control **77**(18), 1567–1579 (2007)
18. Ghods, N., Krstic, M.: Multiagent deployment over a source. IEEE Trans. Control Syst. Technol. 1–10 (2011)
19. Ghods, N., Frihauf, P., Krstic, M.: Multi-agent deployment in the plane using stochastic extremum seeking. In: Proceedings of the Conference on Decision and Control (2010)
20. Ghods, N., Krstić, M.: Speed regulation in steering-based source seeking. Automatica **46**, 452–459 (2010)
21. Ghods, N., Krstić, M.: Source seeking with very slow or drifting sensors. J. Dyn. Syst. Meas. Control **133** (2011)
22. Jakobus, U.: Review of advanced EM modeling techniques in the computer code FEKO. Appl. Comput. Electromagn. Soc. Newsl. **18**(2) (2003)
23. Kanchanavally, S., Ordóñez, R., Schumacher, C.J.: Path planning in three dimensional environment using feedback linearization. In: American Control Conference, Minneapolis, MN (2006)

24. Koksal, M.I., Gazi, V., Fidan, B., Ordóñez, R.: Tracking a maneuvering target with a nonholonomic agent using artificial potentials and sliding mode control. In: 16th Mediterranean Conference on Control and Automation (2008)
25. Krstić, M., Cochran, J.: Extremum seeking for motion optimization: from bacteria to nonholonomic vehicles. In: Proceedings of the Chinese Control and Decision Conference, pp. 18–27 (2008)
26. Lawton, J., Beard, R., Young, B.: A decentralized approach to formation maneuvers. IEEE Trans. Robot. Autom. **19**(6), 933–941 (2003)
27. Leonard, N.E., Fiorelli, E.: Virtual leaders, artificial potentials and coordinated control of groups. In: Proceedings of the IEEE Conf. Decision and Control, Orlando, FL, pp. 2968–2973 (2001)
28. Liu, S.-J., Krstić, M.: Stochastic source seeking for nonholonomic unicycle. IEEE Trans. Autom. Control **46**, 1443–1453 (2010)
29. Mayhew, C.G., Sanfelice, R.G., Teel, A.R.: Robust source-seeking hybrid controller for autonomous vehicles. In: Proceedings of the American Control Conference, pp. 1185–1190 (2007)
30. Mesquita, A.R., Hespanha, J.P., Aström, K.: Optimotaxis: a stochastic multiagent optimization procedure with point measurements. In: Egerstedt, M., Mishra, B. (eds.) Hybrid Systems: Computation and Control. Lecture Notes in Computer Science, vol. 4981, pp. 358–371 (2008)
31. Mitra, A.K., Gates, M., Barber, C., Goodwin, T., Selmic, R., Ordóñez, R., Sekman, A., Malkani, M.: Sensor agnostics for networked MAV applications. In: Evolutionary and Bio-Inspired Computation: Theory and Applications IV. Proc. of SPIE, vol. 7704, p. 77040R (2010)
32. Ordóñez, R., Gates, M., Moma, K., Mitra, A., Selmic, R., Detweiler, P., Cox, C., Parker, G., Goff, Z.: RF emitter localization with position-adaptive MAV platforms. In: IEEE NAECON (2010)
33. Raffard, R.L., Tomlin, C.J., Boyd, S.P.: Distributed optimization for cooperative agents: Application to formation flight. In: Proceedings of the IEEE Conference on Decision and Control, December 2004
34. Reif, J.H., Wang, H.: Social potential fields: a distributed behavioral control for autonomous robots. Robot. Auton. Syst. **27**(3), 171–194 (1999)
35. Rimon, E., Koditschek, D.E.: Exact robot navigation using artificial potential functions. IEEE Trans. Robot. Autom. **8**(5), 501–518 (1992)
36. Rotea, M.A.: Analysis of multivariable extremum seeking algorithms. In: Proceedings of the American Control Conference, vol. 1, pp. 433–437 (2000)
37. Selmic, R.R., Gates, M., Barber, C., Mitra, A., Ordóñez, R.: Position-adaptive direction finding of electromagnetic sources using wireless sensor networks. In: 19th Mediterranean Conference on Control and Automation, June 2011
38. Spooner, J.T., Maggiore, M., Ordóñez, R., Passino, K.M.: Stable Adaptive Control and Estimation for Nonlinear Systems, Neural and Fuzzy Approximator Techniques. Wiley, New York (2002)
39. Stankovic, M.S., Stipanovic, D.M.: Discrete time extremum seeking by autonomous vehicles in a stochastic environment. In: Proceedings of the Conference on Decision and Control, pp. 4541–4546 (2009)
40. Stankovic, M.S., Stipanovic, D.M.: Stochastic extremum seeking with applications to mobile sensor networks. In: Proceedings of the American Control Conference, pp. 5622–5627 (2009)
41. Tan, Y., Nesic, D., Mareels, I., Astolfi, A.: On global extremum seeking in the presence of local extrema. Automatica **45**, 245–251 (2009)
42. Yamaguchi, H.: A cooperative hunting behavior by mobile-robot troops. Int. J. Robot. Res. **18**(8), 931–940 (1999)
43. Yao, J., Ordóñez, R., Gazi, V.: Swarm tracking using artificial potentials and sliding mode control. J. Dyn. Syst. Meas. Control **129**(5), 749–754 (2007)
44. Young, S., Mitra, A.K., Morton, T., Ordóñez, R.: Position-adaptive scatterer localization for radar imaging applications. In: Proc. SPIE, vol. 7308, Orlando, FL, April 2009

45. Zhang, C., Ordóñez, R.: Non-gradient extremum seeking control of feedback linearizable systems with application to ABS design. In: Proceedings of the Conference Decision and Control, pp. 6666–6671 (2006)
46. Zhang, C., Ordóñez, R.: Numerical optimization-based extremum seeking control with application to ABS design. IEEE Trans. Autom. Control **52**(3), 454–467 (2007)
47. Zhang, C., Ordóñez, R.: Robust and adaptive design of numerical optimization-based extremum seeking control. Automatica **45**, 634–646 (2009)
48. Zhang, C., Siranosian, A., Krstić, M.: Extremum seeking for moderately unstable systems and for autonomous target tracking without position measurements. Automatica **43**, 1832–1839 (2007)
49. Zhang, C., Arnold, D., Ghods, N., Siranosian, A., Krstić, M.: Source seeking with nonholonomic unicycle without position measurement and with tuning of forward velocity. Syst. Control Lett. **56**, 245–252 (2007)

Index

C. Zhang, R. Ordóñez, *Extremum-Seeking Control and Applications*,
Advances in Industrial Control,
DOI 10.1007/978-1-4471-2224-1, © Springer-Verlag London Limited 2012

Other titles published in this series (continued):

Soft Sensors for Monitoring and Control of Industrial Processes
Luigi Fortuna, Salvatore Graziani, Alessandro Rizzo and Maria G. Xibilia

Adaptive Voltage Control in Power Systems
Giuseppe Fusco and Mario Russo

Advanced Control of Industrial Processes
Piotr Tatjewski

Process Control Performance Assessment
Andrzej W. Ordys, Damien Uduehi and Michael A. Johnson (Eds.)

Modelling and Analysis of Hybrid Supervisory Systems
Emilia Villani, Paulo E. Miyagi and Robert Valette

Process Control
Jie Bao and Peter L. Lee

Distributed Embedded Control Systems
Matjaž Colnarič, Domen Verber and Wolfgang A. Halang

Precision Motion Control (2nd Ed.)
Tan Kok Kiong, Lee Tong Heng and Huang Sunan

Optimal Control of Wind Energy Systems
Julian Munteanu, Antoneta luliana Bratcu, Nicolaos-Antonio Cutululis and Emil Ceangă

Identification of Continuous-time Models from Sampled Data
Hugues Gamier and Liuping Wang (Eds.)

Model-based Process Supervision
Arun K. Samantaray and Belkacem Bouamama

Diagnosis of Process Nonlinearities and Valve Stiction
M.A.A. Shoukat Choudhury, Sirish L. Shah and Nina F. Thornhill

Magnetic Control of Tokamak Plasmas
Marco Ariola and Alfredo Pironti

Real-time Iterative Learning Control
Jian-Xin Xu, Sanjib K. Panda and Tong H. Lee

Deadlock Resolution in Automated Manufacturing Systems
ZhiWu Li and MengChu Zhou

Model Predictive Control Design and Implementation Using MATLAB®
Liuping Wang

Predictive Functional Control
Jacques Richalet and Donal O'Donovan

Fault-tolerant Flight Control and Guidance Systems
Guillaume Ducard

Fault-tolerant Control Systems
Hassan Noura, Didier Theilliol, Jean-Christophe Ponsart and Abbas Chamseddine

Detection and Diagnosis of Stiction in Control Loops
Mohieddine Jelali and Biao Huans (Eds.)

Stochastic Distribution Control System Design
Lei Guo and Hong Wang

Dry Clutch Control for Automotive Applications
Pietro J. Dolcini, Carlos Canudas-de-Wit and Hubert Béchart

Advanced Control and Supervision of Mineral Processing Plants
Daniel Sbárbaro and René del Villar (Eds.)

Active Braking Control Design for Road Vehicles
Sergio M. Savaresi and Mara Tanelli

Active Control of Flexible Structures
Alberto Cavallo, Giuseppe de Maria, Ciro Natale and Salvatore Pirozzi

Induction Motor Control Design
Riccardo Marino, Patrizio Tomei and Cristiano M. Verrelli

Fractional-order Systems and Controls
Concepcion A. Monje, YangQuan Chen, Blas M. Vinagre, Dingyu Xue and Vincente Feliu

Model Predictive Control of Wastewater Systems
Carlos Ocampo-Martinez

Wastewater Systems
Carlos Ocampo-Martinez

Tandem Cold Metal Rolling Mill Control
John Pitter and Marwan A. Simaan