ACTA UNIVERSITATIS UPSALIENSIS
Uppsala Dissertations from the Faculty of Science and Technology
127

Per Mattsson

Modeling and identification of nonlinear and impulsive systems

UPPSALA
UNIVERSITET

Dissertation presented at Uppsala University to be publicly examined in 2446, ITC, Lägerhyddsvägen 2, Uppsala, Friday, 25 November 2016 at 13:15 for the degree of Doctor of Philosophy. The examination will be conducted in English. Faculty examiner: Professor Roy Smith (Department of Information Technology and Electrical Engineering at the Swiss Federal Institute of Technology (ETH)).

Abstract

Mattsson, P. 2016. Modeling and identification of nonlinear and impulsive systems. *Uppsala Dissertations from the Faculty of Science and Technology* 127. 210 pp. Uppsala: Acta Universitatis Upsaliensis. ISBN 978-91-554-9721-7.

Mathematical modeling of dynamical systems plays a central roll in science and engineering. This thesis is concerned with the process of finding a mathematical model, and it is divided into two parts - one that concentrates on nonlinear system identification and another one where an impulsive model of testosterone regulation is constructed and analyzed.

In the first part of the thesis, a new latent variable framework for identification of a large class of nonlinear models is developed. In this framework, we begin by modeling the errors of a nominal predictor using a flexible stochastic model. The error statistics and the nominal predictor are then identified using the maximum likelihood principle. The resulting optimization problem is tackled using a majorization-minimization approach, resulting in a tuning parameter-free recursive identification method. The proposed method learns parsimonious predictive models. Many popular model structures can be expressed within the framework, and in the thesis it is applied to piecewise ARX models.

In the first part, we also derive a recursive prediction error method based on the Hammerstein model structure. The convergence properties of the method are analyzed by application of the associated differential equation method, and conditions ensuring convergence are given.

In the second part of the thesis, a previously proposed pulse-modulated feedback model of testosterone regulation is extended with infinite-dimensional dynamics, in order to better explain testosterone profiles observed in clinical data. It is then shown how the analysis of oscillating solutions for the finite-dimensional case can be extended to the infinte-dimensional case. A method for blind state estimation in impulsive systems is introduced, with the purpose estimating hormone concentrations that cannot be measured in a non-invasive way. The unknown parameters in the model are identified from clinical data and, finally, a method of incorporating exogenous signals portraying e.g. medical interventions is studied.

Keywords: nonlinear system identification, modeling, impulsive systems, impulse detection

Per Mattsson, Department of Information Technology, Division of Systems and Control, Box 337, Uppsala University, SE-75105 Uppsala, Sweden.

ISSN 1104-2516
ISBN 978-91-554-9721-7
urn:nbn:se:uu:diva-304837 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-304837)

List of papers

This thesis is based on the following papers.

I P. Mattsson, A. Medvedev, and Z. Zhusubaliyev. Pulse-modulated Model of Testosterone Regulation Subject to Exogenous Signals. In *55th IEEE Conference on Decision and Control*, Las Vegas, USA, Dec. 2016

II P. Mattsson, D. Zachariah, and P. Stoica. Recursive nonlinear system identification method using latent variables. *Submitted*, 2016

III P. Mattsson, A. Medvedev, and A. Churilov. Poincaré map for an Impulsive Oscillator with General Hereditary Dynamics. *Submitted*, 2016

IV P. Mattsson, D. Zachariah, and P. Stoica. Recursive identification method for piecewise ARX models: A sparse estimation approach. *IEEE Transactions on Signal Processing*, 64(19):5082–5093, 2016

V P. Mattsson and T. Wigren. Convergence analysis for recursive Hammerstein identification. *Automatica*, 71:179–186, 2016

VI P. Mattsson and A. Medvedev. Modeling of testosterone regulation by pulse-modulated feedback. In *Signal and Image Analysis for Biomedical and Life Sciences*, pages 23–40. Springer, 2015

VII A. Churilov, A. Medvedev, and P. Mattsson. Discrete-time modeling of a hereditary impulsive feedback system. In *53rd IEEE Conference on Decision and Control*, Los Angeles, California, USA, Dec. 2014

VIII A. Churilov, A. Medvedev, and P. Mattsson. Periodical solutions in a pulse-modulated model of endocrine regulation with time-delay. *IEEE Transactions on Automatic Control*, 59(3):728–733, March 2014

IX P. Mattsson and T. Wigren. Recursive identification of Hammerstein models. In *American Control Conference*, June 2014

X P. Mattsson and A. Medvedev. Modeling of testosterone regulation by pulse-modulated feedback: an experimental data study. *AIP Conference Proceedings*, 1559(1), 2013

XI A. Churilov, A. Medvedev, and P. Mattsson. On Finite-dimensional Reducibility of Time-delay Systems under Pulse-modulated Feedback. In *52nd IEEE Conference on Decision and Control*, pages 362–367, 2013

XII P. Mattsson and A. Medvedev. State estimation in linear time-invariant systems with unknown impulsive inputs. In *European Control Conference*, pages 1675–1680, July 2013

XIII A. Churilov, A. Medvedev, and P. Mattsson. Analysis of a pulse-modulated model of endocrine regulation with time-delay. In *51st IEEE Conference on Decision and Control*, pages 362–367, 2012

XIV P. Mattsson and A. Medvedev. Estimation of input impulses by means of continuous finite memory observers. In *American Control Conference*, pages 6769–6774, June 2012

Acknowledgment

I would like to start by thanking my supervisor Professor Alexander Medvedev. You are the one that introduced me to the world of automatic control and system identification. Thank you for providing guidance and knowledge whenever I needed it. I am grateful for the freedom you have given me in my research, and for all the helpful feedback and inspiring new ideas you have provided.

I also want to thank my second supervisor, Professor Petre Stoica, for all the great feedback you have given. You have taught me more than what can be mentioned here, including such diverse topics as estimation and signal processing, optics, bike lights, animal behavior and analysis of Family Guy episodes.

A special thanks also goes out to all my co-authors. Professor Alexander Churilov who, besides being exceptional in finding interesting ways to prove intriguing results, have been a great travel companion and live guidebook during conferences. Professor Zhanybai Zhusubaliyev, who can make the most fantastic bifurcation diagrams. Adj. Professor Torbjörn Wigren, to describe the support you have given me I need a $\otimes$, and I know how you feel about that product. Dave Zachariah who have been a great sounding board and insipiration – since you learned which chair is mine during lunch, it has been awesome working with you!

Thanks to those who gave me feedback during different stages of the manuscript for this thesis: Thomas, Niklas, and Andreas.

And of course, I want to thank everyone at SysCon for creating a persistently exciting (pe) work environment. %*TODO: Figure out someway to express all the nice things I want to say about my current and former colleagues...That they all made it fun to come to the office etc. Maybe something about interesting discussions with Soma – I can more or less copy what she wrote about me. Marcus will probably get annoyed because I ignored some* LaTeX*-warnings* [**?**] *when I compiled this thesis, I should write something about that. Olov fixed my bike several times, but nei-*

ther he nor Daniel mentioned me in their acknowledgments, so might not mention them explicitly. But, yeah, probably write something about the TDB-people, and lunch-friends like Fredrik, Johannes and Egi. And everyone else! I have had so many awesome colleagues during these years, there is no way I can mention all of them. Maybe I should just skip this passage and write "no one mentioned, no one forgotten" instead...
It has been great working, discussing and socializing with all of you – no one mentioned, no one forgotten!

I would also like to thank IK Rex, the greatest cross-country skiing club in the world, and especially Pär Ohlström who encouraged me to start training again, and who also helped me with both ski waxing and rilling (not to be confused with Rill in Chapter 4) while I was doing research (watching TV).

Ylva also deserves a special thanks, for being very supportive, letting me use her car, traveling the world with me and much much more.

I am also grateful to Professor Roy Smith, ETH, for being my faculty opponent, and to Professor Bo Wahlberg, KTH, Associate Professor Maria Prandini, Politecnico di Milano, and Associate Professor Martin Enqvist, Linköping University for agreeing to serve as commite member during the defense of this thesis.

The major part of my PhD has been funded by the European Research Council (Advancded Grant 247035) and the Swedish Research Council (Grant 2012-3153), thank you for that!

Last, but not least, my family. My parents Britt and Björn who have always supported me, no matter what I have decided to do with my life. And my sisters: Frida, Linda, Lill-Anna (she started it! Cf. Acknowledgment in [97]), Karin and Elin – a lone brother cannot wish for a better set of sisters! Special thanks to Linda for the artwork on the cover – in the future I will call you instead of using `plot` in Matlab.

Glossary and notation

Abbreviations

ARMAX	Autoregressive moving average with exogenous input
ARX	Autoregressive with exogenous input
EM	Expectation-maximization
GnRH	Gonadotropin-releasing hormone
i.i.d.	Independent and identically distributed
LASSO	Least absolute shrinkage and selection operator
LH	Lutenizing hormone
LS	Least squares
LTI	Linear time-invariant
MAP	Maximum a posteriori
MIMO	Multiple-input/multiple-output.
ML	Maximum likelihood
MM	Majorization-minimization
NARMAX	Nonlinear ARMAX
NARX	Nonlinear ARX
NMSE	Normalized mean square error
OE	Output error
pe	Persistently exciting
PEM	Prediction error method
PDF	Probability density function
PWARX	Piecewise ARX
RPEM	Recursive PEM
SISO	Single-input/single-output
Te	Testosterone
w.p.1	With probability one
w.r.t	With respect to

Notation

$\mathbb{R}^n, \mathbb{C}^n$	real- and complex-valued n-dimensional vector space
$\mathbb{R}^{n\times m}, \mathbb{C}^{n\times m}$	real- and complex-valued $n\times m$-dimensional matrix space
$n_x, n_u, n_y, n_\theta, \ldots$	dimension of vector $x, u, y, \theta, \ldots$
$A_{i,j}, B_{i,j}, \ldots$	the (i,j)th element of matrix $A, B, \ldots$
$\dot{x}(t)$	time derivative of the signal $x(t)$
I_n	the $n \times n$ identity matrix
$E_{i,j}$	the (i,j)th standard matrix basis, i.e., element (i,j) is 1, and all other elements are equal to 0
q^{-1}	the unit delay operator
$p(x)$	probability density function (PDF) of x
$p(x\|y)$	PDF of x conditioned on y
$\mathrm{E}_x[g(x)]$	expected value with respect to x; $\int g(x)p(x)dx$
$(\cdot)^\top$	vector or matrix transpose
$(\cdot)^*$	the conjugate transpose
$(\cdot)^{-1}$	inverse of matrix
$(\cdot)^\dagger$	Moore-Penrose generalized inverse of a matrix
$\mathrm{tr}(\cdot)$	trace of a matrix
$\mathrm{vec}(\cdot)$	columnwise vectorization of matrix
$\det(\cdot)$	determinant of matrix
$\sigma(\cdot)$	spectrum of a matrix, i.e., set of all eigenvalues
$\|x\|_p$	ℓ_p-norm; $\left(\sum_{i=1}^{n_x} \|x_i\|^p\right)^{1/p}$
$\|x\|_W$	weighted norm; $\sqrt{\mathrm{tr}(x^*Wx)}$, where $W \succ 0$
$\ln(\cdot)$	natural logarithm
$\succ$	generalized inequality; if $A \succ 0$, then A is positive definite, if $A \succ B$ then $A - B$ is positive definite
$\sim$	distributed as
$\approx$	approximately equal to
$\triangleq$	defined as equal to
$\otimes$	Kronecker product
$\odot$	Hadamard product
$H(\cdot)$	the Heaviside step function
$\delta(\cdot)$	the Dirac delta function
$\mathcal{N}(\mu, \Sigma)$	normal distribution with mean μ and covariance matrix Σ
$\mathcal{L}\{\cdot\}$, $\mathcal{L}^{-1}\{\cdot\}$	the Laplace transform, and Laplace inverse transform
$\mathcal{Z}^t$	set of all data collected until time t

Contents

Chapter 1

Introduction

As humans, we use abstract mental models of real-world objects everyday. For example, from past experience, most of us have a mental model of a car, which might say: "Turning the wheel left makes the car go left, pushing down the throttle makes the car go faster, pushing down the brake pedal makes the car go slower, and sounding the horn makes the car go 'beep beep' ". This model of a car says nothing about how an internal combustion engine works, but by just describing how the driver's inputs to the car (steering, throttle, braking, honking) affect the outputs of the car (position, speed and sounds), it is still useful for the driver.

In science and engineering, it is common to use mathematical models, i.e. models described by mathematical equations. Depending on the purpose, the model might describe the internal workings of the object in detail, or it might only give a relation between different inputs and outputs. Before we can use the model we, of course, have to construct it – and this is the main theme of this thesis. The first part of this thesis concerns the matter of how a mathematical model can be identified from experimental data, and in the second part we develop and analyze a mathematical model of testosterone regulation in the human male. Before we dig into the details, this chapter provides a brief history of mathematical modeling, a short overview of important concepts, and ends with an outline of the thesis.

1.1 A (very) brief history of mathematical modeling

Creating abstract models of real-world objects have been a part of human life throughout history. Some very old abstract representations can be seen in cave paintings that were made over 40 000 years ago.

With the development of mathematics, humanity got a powerful tool for abstract thinking, and there is evidence that by 2 000 BC mathematical models were used in at least Babylon, Egypt, and India to improve the quality of everyday life.

Astronomy was one of the first fields where mathematical models were widespread and successfully applied. A milestone was when the Greek philosopher Thales of Miletus was able to predict the solar eclipse of May 28, 585 BC. Around 150 AD, the Greco-Egyptian writer Claudius Ptolemy presented a geocentric model, based on circles and epicycles, that could predict the movement of the sun, moon and planets. Even though the model placed the earth in the center of the solar system, it gave very accurate predictions. Furthermore, the model came with convenient tables that could be used to compute new predictions. For about 1 500 years, the model of Ptolemy remained the (almost) universally accepted model of planetary motion in the solar system.

However, in the 16th century, the heliocentric view started to win recognition thanks to people like Nicolaus Copernicus and Galileo Galilei. In the early 17th century, Johannes Kepler was able to present his three laws of planetary motions. Before we go on, note that the word "law" in this context does not mean "absolute truth". A physical law should rather be understood as a model that has been validated by repeated experiments and observations. Kepler's model of the planetary motion was much simpler and more accurate than the model of Ptolemy, but it did not explain *why* the planets move the way they do. Kepler had basically found his model by painstaking examination of a large data set of astronomical observations, collected by Tycho Brahe. However, for the purpose of predicting the future motion of the planets it was, and still is, a very good model.

Less than a century after Kepler presented his model, Isaac Newton published his three laws of motion. While Kepler's laws concerned the motion of planets, Newton's laws concerned motion in general. He tried to unify all types of motion into three basic principles. Newton's laws are still widely used, but we now know that they are not valid in certain domains that Newton could not observe with the technology available at the time. Einstein's special relativity shows that Newton's laws do not hold at very high speeds, and quantum mechanics show that they are not valid at very small scales. However, for objects we observe in our everyday life, Newton's laws give a good model for how they move. Newton also showed how his three basic principles implied Kepler's laws, and thus, in a sense, explain why planets move as they do.

Before we end this history lesson, we have to mention Carl Friedrich Gauss. Newton discovered the laws that governed the motion of celestial bodies, but to actually apply them to a real-world planet, we have to make observations in order to compute some unknown parameters. The

problem is that, due to imperfect instruments, these observations always contain measurement errors. When the planet Ceres was discovered in 1801, Gauss was able to use observations in order to compute its orbit. He laid out his methods in [51], where he explained that he used a least-squares criterion to locate the orbit that best fit the observations. He also justified his method by a theory of errors, that led him to a derivation of the normal distribution, which today often is refereed to as the Gaussian distribution. As will be seen in Section 2.4, these methods are still used in statistics and system identification to this day.

1.2 Dynamical systems

In this thesis, a system typically refers to a real-world object that we would like to model. For example, the system could be an airplane. The airplane is controlled through the elevator position and the engine thrust, which we call the inputs to the system. The inputs at time t are denoted $u(t)$. We might then measure the altitude and speed of the airplane, and we call these the outputs of the system. The outputs at time t are denoted $y(t)$. Hence, a model of a system could be a description of how the inputs $u(t)$ affects the outputs $y(t)$. However, the airplane will also be affected by atmospheric conditions etc. that the pilot cannot control – we usually view such variables as disturbances. Preferably, we would like our model to take into account both the effect of inputs manipulated by the pilot and the disturbances.

In general, a system can be illustrated in a block-diagram as in Figure 1.1, where the box represents a system $\mathcal{S}$ that is driven by the external input signal $u(t)$ and the disturbances $v(t)$ to produce the output signal $y(t)$. When the output $y(t)$ at time t only depends on the inputs and disturbances at time t, then we say that the system is static. However, in most real-world objects, the output at time t will depend on past inputs and disturbances too, and in such case we say that the system is dynamical.

In our everyday life, we encounter dynamical systems all the time, such as industrial robots, airplanes, cars, etc. We can also view the human body as a complex dynamical system with many interacting subsystems, or, as in Part II of this thesis, view the testosterone regulation in the human male as a dynamical system.

1.3 Modeling principles

In this thesis, we are concerned with finding a mathematical model for the box in Figure 1.1. Depending on how we view this box, and our

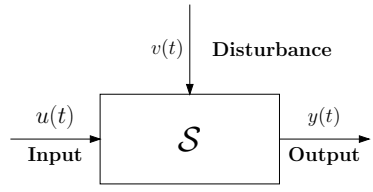


Figure 1.1: A dynamical system $\mathcal{S}$, with input $u(t)$, output $y(t)$ and disturbance $v(t)$. Here t denotes time.

prior knowledge about the system it represents, different strategies can be used when constructing the model.

White-box modeling
In this approach, we create our model from first principles, such as Newton's laws, Maxwell's equations, or biological properties of the system. In this way, the model can give us a deeper understanding of how the physical system actually works, as was the case when Newton derived the model of planetary motion from his basic laws of general motion. However, deriving such a model for complex systems might be very time-consuming or almost impossible, so different simplifying assumptions often have to be used. For this reason it is common that the model is only valid in very idealized situations.

Black-box modeling
This is arguably the most common approach in system identification, and it is also the approach that will be taken in Part I of this thesis. Here $\mathcal{S}$ in Figure 1.1 is viewed as a completely unknown black-box, and we construct a model by first observing the system through measurements of the inputs and outputs. Then we construct the model by fitting it to the observed data in some way. In a sense, this is how Kepler found his laws, i.e. by studying empirical data (however, Kepler himself might not have agreed with this statement). For Kepler, it took years to go through all the data, but with a modern computer and methods of system identification this process can often be completed within minutes. While a model obtained in this way might not give any direct physical insights, it can still be useful, e.g. for making predictions. A drawback of black-box models is that they are usually only valid under conditions similar to when the identification data were collected – just as Newton's laws are only valid for the speeds and scales that he could observe with the technology available at the time.

Shades of gray
The two approaches mentioned above should be seen as two extremes in a spectrum of modeling principles. We might not be able to construct

a full model using only our prior knowledge of the system, as in the white-box approach, but we might still have some physical insight that can be combined with examination of empirically measured data. As an example, Gauss had insights from Newton's laws when he found the orbit of Ceres, but he also had to use observations in order to estimate some unknown parameters. In Part II of this thesis, we study a previously proposed basic model of testosterone regulation in the human male that is motivated by biological properties of the endocrine system. An examination of clinical data then leads to an extension of the model, and estimates of the unknown parameters. Creating a model in this way lies somewhere between white-box and black-box modeling, and is thus often referred to as gray-box modeling. Gray-box modeling is a very broad group of strategies that can be further divided into different shades of gray as in, e.g. [90].

1.4 What is the purpose of the model?

A well known quote in modeling is

> "Essentially, all models are wrong, but some are useful",

usually attributed to George E. P. Box [18]. In order to know if a model is useful, we must first ask ourself what the purpose of the model is. Three common uses of models are:

- *Simulation*
  Given a model of the system under study, we can simulate it in order to study how it responds to different inputs etc., by using a computer. Since simulation in a computer is typically a lot cheaper than experiments on the real system, this can be a cost-effective way to learn more about the system.
- *Predictions*
  If we measure the current input and output of the system, the model can often be used to predict how the system will behave in the future. If we combine this with system identification, we get a method for using data measured in the past in order to predict future outcomes. Mathematical models have been used for prediction in many different applications, such as weather forecasts [95], healthcare [81], economics [63], and predicting which movies you will like [11].
- *Automatic control*
  The goal in automatic control is to get a dynamical system $\mathcal{S}$ to behave in a desired way, without direct human intervention. This is typically done through a feedback control loop like in Figure 1.2.

Here the user specifies a reference signal that the system should follow, and the controller $\mathcal{C}$ automatically chooses an appropriate input signal based on the reference signal and the current output. When constructing the controller, it is useful to have a model of the system $\mathcal{S}$. For linear models, there is a vast theory about constructing suitable controllers [54]. One popular control strategy, that is also applicable to nonlinear models, is model predictive control. In this strategy, the model is used to predict how the system will behave in the future for a given input, and the input that results in the best behavior is chosen.

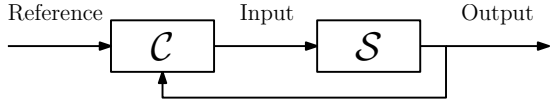


Figure 1.2: A typical feedback control system.

1.5 The principle of parsimony

John von Neumann has been attributed with the following famous quote:

> "With four parameters I can fit an elephant, and with five I can make him wiggle his trunk."

That this is indeed possible was later shown in [108]. The point of von Neumann's quote is that, with a flexible enough model structure, we can fit any measured data – but this does not necessarily make it a good model. The problem here is that an excessively complicated model can overfit the data by describing random disturbances included in the specific data set used. Such a model will typically not be reliable when confronted with data that was not used during the modeling. So, if two models explain all of our measured data equally well, which one is better?

The parsimony principle is used in all of science. It basically says that, if there are two solutions to a problem, then the simpler solution is better. During the 1960's, the U.S. Navy succinctly phrased this principle as "KISS – Keep it simple, stupid!". According to this principle, the answer to the above question is that the less complicated model is better. This can be compared to Section 1.1, where we saw that Kepler's model was less complicated, but still superior, compared to the model of Ptolemy.

The parsimony principle can be seen as a useful heuristic, but in system identification and statistics it can also often be justified by mathematics [135, 70].

1.6 Modeling concepts

When designing a mathematical model, there are some core concepts that are useful to keep in mind. In this section, we discuss some of the concepts that will be used throughout this thesis.

Casual – Non-casual
In a causal model, the output at time t only depends on past inputs, disturbances, and outputs. A model where the output at time t can depend on future events is called *non-causal*. Since the real-world is often considered to be a causal system, mathematical models of real world objects tend to be causal. However, there might be cases where a non-causal model is useful. For example, in insider trading on the stock market, the insider has information about future events.

Time-invariant – Time-variant
In a time-invariant model, the dynamics does not depend on explicitly on time. Intuitively, this means that, if the input $u(t)$ and distrubances $v(t)$ produce the output $y(t)$, then time-shifted signals $u(t+\tau)$ and $v(t+\tau)$ result in a time-shifted output $y(t+\tau)$. Many systems can be well described by time-invariant models, since the behavior of the system does not depend on which exact time the experiment started.

In other systems, however, the time the experiment starts might be very important. In, e.g., biomedicine, the circadian rhythm has the effect that we might obtain different results depending on which time of day a drug is administered. For these cases time-variant models may be needed.

Continuous time – Discrete time
In a continuous-time model, the time variable t ranges over the entire real number line, and between any two points in time a signal, e.g. the output $y(t)$, can take an infinite number of different values. This is how we typically view the real world, that things change continuously.

In discrete time, the variable t ranges over a discrete set of "points in time". In some cases, it is natural to model the real world using discrete time. For example, if the output $y(t)$ of the model is an estimate of the annual gross domestic product (GDP), it is natural to let t denote the year, so that

$$t \in \{\ldots, 1929, \ldots, 1973, \ldots, 1990, \ldots, 2008, \ldots\}.$$

It also common to use discrete time models even though the underlying system is considered to change continuously in time. One reason for this is that many real-world systems are controlled by a digital computer, which works in discrete time. Furthermore, we can only sample the

signals of the system at discrete points in time anyway, so we might consider models that describe how the state of the system changes from one sampling time to the next one.

Hybrid and impulsive models

While it is common to model a system either in continuous or discrete time, there is growing interest in hybrid models that exhibit both continuous and discrete dynamic behaviors, see for example [55]. This can be useful, for example, when the system consists of both analog and digital components.

Another opportunity for hybrid models arises when changes in the system at hand occur at dramatically different rates. An example of this is the gonadotropin-releasing hormone in the human, which is released in bursts from the hypothalamus of the brain. Since this type of release is so much faster than the dynamics of other hormones, we can model the bursts as instantaneous impulses that occur at discrete points in time. This results in a model where continuous-time equations are used between the impulsive events, and a different set of equations is used to describe how the state of the system instantaneously changes when an impulse occurs. The subclass of hybrid models that make use of impulses in this way is typically called impulsive models [58].

Linear – Nonlinear

A model is linear if it is described by a linear operator. Intuitively, this means that, if the input $u_1(t)$ gives the output $y_1(t)$ and the input $u_2(t)$ gives $y_2(t)$, then the input $u(t) = a_1u_1(t) + a_2u_2(t)$ gives the output $y(t) = a_1y_1(t) + a_2y_2(t)$. Real-world systems are virtually never linear, but it is often possible to approximate their dynamics as linear around an operation point. Linear models constitute a restricted and well-defined class of models, for which many attractive properties have been derived.

A nonlinear model, on the other hand, is a model that is not linear. In fact, the term "nonlinear" in practice means "not necessarily linear". So, assuming that a system is nonlinear is rather saying that we do not assume anything. This is, of course, problematic if we want to construct general theories about nonlinear models – with no assumptions at all, we cannot say much. However, the title of this thesis clearly states that it concerns modeling and identification of nonlinear systems. The way we deal with the aforementioned problem is that we, as it is often done in the literature on nonlinear modeling, consider restricted and well-defined subclasses of nonlinear models that can still express a wide range of different dynamical behaviors.

1.7 Outline and contributions

This thesis is divided into two parts. In the first part, we introduce the field of nonlinear system identification and then go on to develop new recursive identification methods for some popular nonlinear model structures. Part II concerns modeling of testosterone regulation in the human male. In this part, we take a gray-box modeling approach, and extend a previously proposed impulsive model with infinite-dimensional dynamics. The resulting closed-loop hybrid model is analyzed in detail and unknown parameters are estimated from clinical data using techniques from system identification.

The main contributions of this thesis appear in Chapters 3-5 and Chapters 7-10. Below a brief summary of each chapter is provided.

Part I

Chapter 2: Introduction to system identification

In this chapter, a brief overview of nonlinear system identification is given, with a focus on results that will be utilized in the rest of this thesis.

Chapter 3: Identification of nonlinear models using latent variables

In this chapter, a method for identifying a nonlinear system with multiple inputs and outputs is presented. We develop a latent variable framework in order to model the errors of a nominal predictor. Using the maximum likelihood principle, we derive a criterion for identifying both the nominal model and the error statistics. The resulting optimization problem is tackled using a majorization-minimization approach. This approach leads to the optimization of a convex criterion, and a recursive algorithm that solves the optimization problem is derived. The error statistics can finally be used to refine the nominal predictor. The method finds parsimonious refined predictor models and is tested on both synthetic and real-life nonlinear systems. The chapter is based on the material in:

- P. Mattsson, D. Zachariah, and P. Stoica. Recursive nonlinear system identification method using latent variables. *Submitted*, 2016.

Chapter 4: Identification of PWARX models

This chapter deals with identification of nonlinear systems using piecewise linear models. By a sparse overparameterization, this challenging

problem is turned into a convex optimization problem. The chapter is based on the material in

- P. Mattsson, D. Zachariah, and P. Stoica. Recursive identification method for piecewise ARX models: A sparse estimation approach. *IEEE Transactions on Signal Processing*, 64(19):5082–5093, 2016,

but extends the results therein to the multi-input/multi-output case, and shows that the method is connected with the framework developed in Chapter 3.

Chapter 5: Identification of Hammerstein models

In this chapter, a recursive prediction error method based on the Hammerstein model structure is derived. The convergence properties of the algorithm are analyzed by application of the associated differential equation method, and conditions ensuring convergences are given. The chapter draws upon material contained in:

- P. Mattsson and T. Wigren. Convergence analysis for recursive Hammerstein identification. *Automatica*, 71:179–186, 2016.
- P. Mattsson and T. Wigren. Recursive identification of Hammerstein models. In *American Control Conference*, June 2014 .

Part II

Chapter 6: Modeling of endocrine systems

This chapter gives a brief introduction to endocrine systems, with a focus on testosterone regulation. It also introduces mathematical models that have been used in the literature.

Chapter 7: Pulse-modulated feedback

This chapter starts with a short introduction to pulse-modulated systems. Then a broad class of impulsive models is introduced, where a linear hereditary plant with a cascade structure operates under intrinsic impulsive feedback. The plant incorporates a distinct infinite-dimensional block, which is modelled using finite-memory pseudodifferential operators. This class of operators includes, among others, both pointwise and distributed time-delays. A method for deriving Poincaré maps that capture the propagation of the continuous dynamics between impulses in the pulse-modulated feedback is proposed. These Poincaré maps can then be used in order to analyze periodical solutions to the system equations. This chapter is based on material in:

- P. Mattsson, A. Medvedev, and A. Churilov. Poincaré map for an Impulsive Oscillator with General Hereditary Dynamics. *Submitted*, 2016.

- A. Churilov, A. Medvedev, and P. Mattsson. Periodical solutions in a pulse-modulated model of endocrine regulation with time-delay. *IEEE Transactions on Automatic Control*, 59(3):728–733, March 2014.
- A. Churilov, A. Medvedev, and P. Mattsson. Discrete-time modeling of a hereditary impulsive feedback system. In *53rd IEEE Conference on Decision and Control*, Los Angeles, California, USA, Dec. 2014.
- A. Churilov, A. Medvedev, and P. Mattsson. On Finite-dimensional Reducibility of Time-delay Systems under Pulse-modulated Feedback. In *52nd IEEE Conference on Decision and Control*, pages 362–367, 2013.
- A. Churilov, A. Medvedev, and P. Mattsson. Analysis of a pulse-modulated model of endocrine regulation with time-delay. In *51st IEEE Conference on Decision and Control*, pages 362–367, 2012.

Chapter 8: Blind estimation in impulsive systems

This chapter deals with continuous-time linear models subject to unknown impulsive inputs. This type of models arises e.g. in the context of hybrid systems with intrinsic pulse-modulated feedback, such as the model considered in Chapter 9. In order to estimate the timing and weights of the impulses, a method making use of a continuous finite-memory convolution operator is suggested. Furthermore, it is also shown how this operator can be used together with an impulsive observer in order to perform blind state estimation. This chapter is based on the material in:

- P. Mattsson and A. Medvedev. State estimation in linear time-invariant systems with unknown impulsive inputs. In *European Control Conference*, pages 1675–1680, July 2013.
- P. Mattsson and A. Medvedev. Estimation of input impulses by means of continuous finite memory observers. In *American Control Conference*, pages 6769–6774, June 2012.

Chapter 9: Identification of the testosterone model

In this chapter, the pulse-modulated model of testosterone feedback regulation introduced in Chapter 6 is extended with infinite-dimensional dynamics, to better explain the testosterone profiles observed in clinical data. The parameters in the model are then estimated from hormone concentrations measured in human males, and simulation results from the full closed-loop system are provided. The chapter draws upon material found in:

- P. Mattsson and A. Medvedev. Modeling of testosterone regulation by pulse-modulated feedback. In *Signal and Image Analysis for Biomedical and Life Sciences*, pages 23–40. Springer, 2015.
- P. Mattsson and A. Medvedev. Modeling of testosterone regulation by pulse-modulated feedback: an experimental data study. *AIP Conference Proceedings*, 1559(1), 2013.

Chapter 10: Modeling of exogenous signals
In this chapter, the effect of continuous exogenous signals acting on the pulse-modulated model in Chapter 9 is studied. In the context of endocrine systems, the exogenous signals represent e.g. the drugs used in hormone replacement therapies, or the effects due to the carcadian rhythm, and interactions with other endocrine loops in the organism. It is demonstrated that the endocrine loop with an exogenous signal entering the continuous part can be equivalently described by proper modifications in the pulse-modulation functions of the autonomous model. This extends the scope of the autonomous pulse-modulated models of endocrine regulation studied in Chapter 7 to a broader class of problems, such as therapy optimization, where an exogenous signal is involved. The cases of constant and harmonic exogenous signals are treated in detail and illsutrated by bifurcation analysis. The chapter draw upon material found in:

- P. Mattsson, A. Medvedev, and Z. Zhusubaliyev. Pulse-modulated Model of Testosterone Regulation Subject to Exogenous Signals. In *55th IEEE Conference on Decision and Control*, Las Vegas, USA, Dec. 2016.

Part I:
Nonlinear system identification

Chapter 2

Introduction to system identification

In this chapter, the field of system identification is briefly reviewed. System identification has been an active field of research for more than half a century, and has resulted in many theoretical insights and useful algorithms. This chapter will by no means try to cover all parts of system identification. For a more extensive introduction to system identification, there are many excellent text books, such as [135], [88], [91] and [123].

2.1 Introduction

System identification is about devising mathematical models of dynamical systems from experimental data. From a dynamical system, such as in Figure 1.1, we can collect data by measuring the inputs $u(t)$ and outputs $y(t)$. The data are typically measured at discrete times $t = 1, \ldots, N$, and we denote all data collected until time t as

$$\mathcal{Z}^t = \{(y(1), u(1)), (y(2), u(2)), \ldots, (y(t), u(t))\}\,.$$

It is often possible to set up the experimental conditions in such a way that the measured data give us as much information as possible about the system under study, e.g. by choosing the inputs $u(t)$ in a suitable way. This is an important part of the system identification method.

When the data have been collected, the goal of system identification is to construct a mathematical model of the system. This is often done by considering a specific collection of mathematical models and, based on some criterion or inference rules, picking out the model that fits the observed data best. Finally, we should always verify that the model we have found is suitable for its intended use. This process is called model validation.

The system identification method can be summarized as follows:

1. Design an experiment, run it on the system, and collect data.
2. Choose a model structure, i.e., a set, or collection, of mathematical models. We denote this $\mathcal{M}$.
3. Use the collected data to select a model from $\mathcal{M}$.
4. Validate the model. This is the process of ensuring that the model chosen in Step 3 is valid for other data sets.

The above steps should be seen as an iterative scheme. If the obtained model is not good enough, a different model structure might do a better job, or maybe we have to revisit the experiment design to gather more informative measurements.

Of course, there are several ways of carrying out each of these steps, and this is one of the reasons why system identification offers such a myriad of methods. In Section 2.2, we shortly discuss Step 1. In Section 2.3 a (quite) general framework for nonlinear model structures is introduced and some popular model structures, that can be used in Step 2 above, are described. Section 2.4-2.5 consider Step 3, and finally Step 4 is briefly discussed in Section 2.6.

2.2 Experimental setup

Before we even start to collect data from a system, it is a good idea to think about the experimental conditions. In many cases, the user can influence what type of input signal to use, and how often measurements should be taken. Both of these choices can have profound effect on the system identification process. The design of experimental conditions is a research field on its own right, and we will only touch upon it briefly in this thesis.

As mentioned in Section 1.3, a model identified from measured data is typically only valid under similar conditions. Hence, one guideline in design of the input signal is to ensure that it covers the situations that we are interested in. When the input signal is chosen, it is also important to consider the type of model structures we will use for identification, since the experiment should be informative enough to to discriminate between the models in the model structure [88].

In linear system identification, the importance of input signals that are exciting enough in frequency is well known [135]. For nonlinear system identification, the problem of choosing a suitable input signal is much more involved, since the behavior of the system also changes depending on the amplitude of the signals. In summary, it can be said that we can only hope to find an accurate model at the frequencies and amplitudes that are in the data set used for identification.

2.3 Model structures

Elephants are apparently popular in proverbs, cf. Section 1.5. A saying, often attributed to Stan Ulman and relevant to nonlinear modeling, is that this field is as huge as "non-elephant" zoology. In other words, describing all possible nonlinear model structures is a next to impossible task, as discussed in Section 1.3.

Despite this fact, we study a relatively general framework of nonlinear discrete time dynamical models in Section 2.3.1. In this framework, the model is considered to be a predictor that can predict future outputs given measurements of past inputs and outputs. While this framework has been proved useful in deriving new identification methods, it is by no means the only way in which we can study nonlinear models, cf. [59].

In Section 2.3.2, we discuss how the elephant in the room (linear models) fits into the framework of Section 2.3.1, and in Section 2.3.3 some popular nonlinear model structures are presented.

2.3.1 Predictor models

In a causal dynamical system $\mathcal{S}$, the output $y(t)$ only depends on what has happened before time t. From the user's perspective, the knowledge about the past is in the measured data $\mathcal{Z}^{t-1}$.

If we let $\hat{y}(t)$ be the output of our model, it thus seems reasonable to consider a discrete time model of the system to be a function of $\mathcal{Z}^{t-1}$, i.e.

$$\hat{y}(t) = g(\mathcal{Z}^{t-1}, t). \tag{2.1}$$

Here $\hat{y}(t)$ can be seen as the output we expect, or predict, at time t given the data collected until time $t-1$. For this reason, a model like (2.1) is often called a *predictor model* [88].

For the rest of this chapter, we will concentrate on time-invariant models, which means that we consider models where $g(\cdot)$ does not depend explicitly on the time t.

Due to modeling errors and unmeasured disturbances, the predicted output $\hat{y}(t)$ does, in practice, not coincide with the actual output $y(t)$. Therefore, we also introduce the *prediction error* $\varepsilon(t) = y(t) - \hat{y}(t)$, and note that the output $y(t)$ can be written as

$$y(t) = \hat{y}(t) + \varepsilon(t) = g(\mathcal{Z}^{t-1}) + \varepsilon(t). \tag{2.2}$$

Hence, any function $g(\cdot)$ of past data can be considered to be a model of $\mathcal{S}$. However, for the model to be of interest, the prediction error $\varepsilon(t)$ should be small in some sense, so that $\hat{y}(t)$ can be seen as a good prediction of $y(t)$. The problem of determining weather a given model is good enough is briefly discussed in Section 2.6.

Stochastic models

In the previous section, we just stated that the prediction errors are given by $\varepsilon(t) = y(t) - \hat{y}(t)$. To get a complete model of the system, we might also try to model these prediction errors.

If we run the same experiment on a real-world system two times with exactly the same input signal, we will usually get different prediction errors. This can be due to measurement noise, and other disturbances that affect the system. It is common to model these disturbances by considering the prediction errors as a stochastic process.

In the framework used here, this can be done by assuming a conditional probability density function for $\varepsilon(t)$,

$$\varepsilon(t)|\mathcal{Z}^{t-1} \sim p_\varepsilon(\varepsilon(t)|\mathcal{Z}^{t-1}).$$

This relationship gives us the distribution of $\varepsilon(t)$ given all data measured until time $t-1$. However, note that this is *a model* for the prediction errors, i.e. an assumption about how the prediction errors are distributed, and the true prediction errors might not follow this distribution.

In summary, we can write the *complete stochastic model* as

$$\hat{y}(t) = g(\mathcal{Z}^{t-1}), \tag{2.3}$$
$$\varepsilon(t)|\mathcal{Z}^{t-1} \sim p_\varepsilon(\varepsilon(t)|\mathcal{Z}^{t-1}), \tag{2.4}$$
$$y(t) = \hat{y}(t) + \varepsilon(t). \tag{2.5}$$

The model structure $\mathcal{M}$ then contains a set of models on the form (2.3)-(2.4). From (2.5), it can be seen that the distribution of the output $y(t)$, according to the model, is given by [88]

$$p(y(t)|\mathcal{Z}^{t-1}) = p_\varepsilon(y(t) - \hat{y}(t)|\mathcal{Z}^{t-1}). \tag{2.6}$$

A common assumption on the prediction errors is that they are independent and identically distributed (i.i.d.). In this case,

$$p_\varepsilon(\varepsilon(t)|\mathcal{Z}^{t-1}) = p_\varepsilon(\varepsilon(t)),$$

i.e. the prediction errors $\varepsilon(t)$ are independent of the data $\mathcal{Z}^{t-1}$. Intuitively, this would be the case when our model $g(\mathcal{Z}^{t-1})$ captures all the information in $\mathcal{Z}^{t-1}$ that can be used to predict the value of $y(t)$, and $\varepsilon(t)$ only contains random disturbances that are independent from previous measurements.

Parametric models

To find the function $g(\cdot)$ among all possible functions from a finite data set is in general an intractable problem, so we have to restrict our search to some family of functions. Here we concentrate on so-called parametric

models, where the function $g(\cdot)$ is parameterized by a finite vector of parameters θ. Sometimes, we explicitly write out the dependence on θ as

$$\hat{y}(t|\theta) = g(\mathcal{Z}^{t-1}, \theta), \tag{2.7}$$

$$\varepsilon(t,\theta)|\mathcal{Z}^{t-1}, \theta \sim p_\varepsilon(\varepsilon(t,\theta)|\mathcal{Z}^{t-1}, \theta). \tag{2.8}$$

Finally, the parameter vector θ might also be restricted to belong to a certain set $\mathcal{D}_\mathcal{M} \subset \mathbb{R}^{n_\theta}$, where n_θ is the dimension of θ. For example, the model structure might only include parameter vectors for which the model is stable. In Sections 2.3.2-2.3.3 different ways to parametrize $g(\mathcal{Z}^{t-1}, \theta)$ are discussed.

Bayesian models

A common interpretation of probabilities is as the frequency of random, repeatable events. This is called the frequentist interpretation. In contrast to this, the Bayesian interpretation, see e.g. [122], perceives probabilities as quantification of uncertainty. Hence, $p_\varepsilon(\varepsilon(t,\theta)|\mathcal{Z}^{t-1}, \theta)$ in (2.8) is a measure of our uncertainty about $\varepsilon(t,\theta)$ if we already knew the parameter vector θ and the data $\mathcal{Z}^{t-1}$.

In system identification, we are uncertain about the parameter vector θ. Hence, in a Bayesian framework, θ is viewed as a random variable with some probability distribution $p(\theta)$. This distribution is called the prior distribution, since it encodes our uncertainty about θ prior to any observation of data. In order to get a Bayesian model structure, we thus add a model of our prior uncertainty about θ to (2.7)-(2.8) as

$$\theta \sim p(\theta). \tag{2.9}$$

When data are observed, we can then infer information about θ and thus update our uncertainty. This inference can formally be done using Bayes rule,

$$p(\theta|\mathcal{Z}^N) = \frac{p(\mathcal{Z}^N|\theta)p(\theta)}{p(\mathcal{Z}^N)},$$

as will be seen in Section 2.4.2.

Latent variables

Many popular model structures introduce a latent variable $x(t) \in \mathbb{R}^{n_x}$, that is not directly observed. This latent variable can be used to describe an internal state of the model. The latent variable is typically related to the measured data through some relationship like

$$x(t+1) = f(x(t), y(t), u(t)). \tag{2.10}$$

It is then assumed that the function $g(\cdot)$ can be computed using only the finite-dimensional latent variable $x(t)$ and the current input $u(t)$, i.e.

$$\hat{y}(t) = g(x(t), u(t)). \tag{2.11}$$

Since $x(t)$ contains the information needed for predicting $\hat{y}(t)$, it is often called the state vector. A model with latent variables can often be more concise then a model based directly on input-output data, but introduces the problem that the latent variable cannot be observed directly.

2.3.2 Linear models

Linear time-invariant (LTI) models are among the most commonly used models in control theory and system identification, and are described in many textbooks, e.g. [75, 128]. In this section, we concentrate on single-input/single-output (SISO) models, but extensions to multiple-input/multiple-output (MIMO) systems are possible.

A predictor model (2.3)-(2.5) is a linear model if $g(\mathcal{Z}^{t-1})$ is a linear function of $\mathcal{Z}^{t-1}$. Furthermore, in linear models, the prediction error $\varepsilon(t)$ is typically modeled as a white noise process with zero mean. There are several ways to parameterize a linear model, and a few of them are mentioned below.

Input-output models

First assume that $g(\mathcal{Z}^{t-1})$ is linear and depends on the n_a most recent outputs and the n_b most recent inputs, i.e.,

$$\hat{y}(t) = -\sum_{i=1}^{n_a} a_i y(t-i) + \sum_{i=1}^{n_b} b_i u(t-i). \tag{2.12}$$

Inserting this into (2.5), we get

$$y(t) = -\sum_{i=1}^{n_a} a_i y(t-i) + \sum_{i=1}^{n_b} b_i u(t-i) + \varepsilon(t). \tag{2.13}$$

Introduce the forward shift operator q, so that for all t we get $qy(t) = y(t+1)$ and, correspondingly, $q^{-1}y(t) = y(t-1)$. Then (2.13) can be rewritten in the compact form

$$A(q)y(t) = B(q)u(t) + \varepsilon(t), \tag{2.14}$$

where

$$A(q) = 1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a}, \tag{2.15}$$

$$B(q) = b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b}. \tag{2.16}$$

The model in (2.14) is called an ARX model, where AR refers the autoregressive part $A(q)y(t)$, and X refers to the exogenous input $B(q)u(t)$. In the ARX-model, there are a finite number of parameters, but a drawback with the predictor in (2.12) is that it only uses the most recent inputs and outputs in order to compute $\hat{y}(t)$.

In general, $\hat{y}(t)$ could be a function of all past inputs and outputs. In order to incorporate all past data with a finite number of parameters, we can compute $\hat{y}(t)$ recursively as

$$\hat{y}(t) = -\sum_{i=1}^{n_a} c_i \hat{y}(t-i) + \sum_{i=1}^{n_a} d_i y(t-i) + \sum_{i=1}^{n_b} b_i u(t-i). \tag{2.17}$$

In order to ensure that this recursion is stable, we assume that all solutions to

$$z^{n_a} + \sum_{i=1}^{n_a} c_i z^{n_a - i} = 0$$

lie inside the unit circle. For more about stability in stochastic dynamical systems, see e.g. [134]. Noting that $\hat{y}(t) = y(t) - \varepsilon(t)$, we can rewrite (2.17) as

$$y(t) = -\sum_{i=1}^{n_a} a_i y(t-i) + \sum_{i=1}^{n_b} b_i u(t-i) + \varepsilon(t) + \sum_{i=1}^{n_a} c_i \varepsilon(t-i),$$

where $a_i = c_i - d_i$, or with the shift operator q as

$$A(q)y(t) = B(q)u(t) + C(q)\varepsilon(t), \tag{2.18}$$

where $A(q)$ and $B(q)$ are defined as in (2.15)-(2.16) and

$$C(q) = 1 + c_1 q^{-1} + \cdots + c_{n_a} q^{-n_a}.$$

The model in (2.18) is referred to as an ARMAX-model, since it extends the ARX-model with the moving average $C(q)\varepsilon(t)$. Also note that, if $C(q) = 1$, then (2.18) becomes an ARX-model. Another important special case is when $A(q) = C(q)$, so that $d_i = 0$ for $i = 1, \ldots, n_a$, in which case (2.18) is refereed to as an output error (OE) model. Note that, for an OE-model, the predicted output $\hat{y}(t)$ does not depend on the actual output $y(t)$, so with such a model predictions can be made using only the input signal $u(t)$.

Regression models

Linear models can be formulated in many different equivalent ways. For example, the predictor for the ARX model in (2.12) can be written as

$$\hat{y}(t|\theta) = \varphi^\top(t)\theta, \tag{2.19}$$

where

$$\theta = \begin{bmatrix} a_1 & \cdots & a_{n_a} & b_1 & \cdots & b_{n_b} \end{bmatrix}^\top,$$
$$\varphi(t) = \begin{bmatrix} -y(t-1) & \cdots & -y(t-n_a) & u(t-1) & \cdots & u(t-n_b) \end{bmatrix}^\top.$$

A model on the form (2.19) is called a linear regression model, since it is linear in the parameters θ. Such models are widely studied in statistics.

The predictor for the ARMAX model in (2.17) can similarly be written as

$$\hat{y}(t|\theta) = \varphi^\top(t,\theta)\theta, \tag{2.20}$$

where

$$\theta = \begin{bmatrix} a_1 & \cdots & a_{n_a} & b_1 & \cdots & b_{n_b} & c_1 & \cdots & c_{n_a} \end{bmatrix}^\top, \tag{2.21}$$
$$\varphi(t,\theta) = [-y(t-1) \quad \cdots \quad -y(t-n_a) \quad u(t-1) \quad \cdots \quad u(t-n_b)$$
$$\varepsilon(t-1,\theta) \quad \cdots \quad \varepsilon(t-n_a,\theta)]^\top, \tag{2.22}$$

and $\varepsilon(t-i,\theta) = y(t-i) - \hat{y}(t-i|\theta)$. Even though (2.20) looks similar to the linear regression in (2.19), it is in general nonlinear in the parameters since we need θ in order to compute $\varphi(t,\theta)$. A model on the form (2.20) is often called pseudolinear regression [91].

State-space models

Now consider using latent variables as in (2.10)-(2.11). Assuming that both $f(\cdot)$ and $g(\cdot)$ are linear in their arguments, we obtain

$$\begin{aligned} x(t+1) &= Fx(t) + Gy(t) + Hu(t), \\ \hat{y}(t) &= Cx(t) + Du(t), \end{aligned}$$

for some matrices F, G, H, C and D. Inserting $y(t) = \hat{y}(t) + \varepsilon(t)$, this can be rewritten as

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + w(t), \\ y(t) &= Cx(t) + Du(t) + \varepsilon(t), \end{aligned} \tag{2.23}$$

where

$$A = F + GC, \quad B = H + GD, \quad w(t) = G\varepsilon(t).$$

A model on the form of (2.23) is referred to as a state-space model.

2.3.3 Nonlinear models

While linear models are well-studied and often used in practical applications, there are many occasions when they do not give a good enough representation of the dynamical system of interest. For this reason, we now turn to nonlinear models.

Volterra models

In the 1880's, the Italian mathematician Volterra used the representation

$$\hat{y}(t) = \sum_{k=1}^{\infty} \int_0^{\infty} \cdots \int_0^{\infty} h_k(\tau_1, \ldots, \tau_k) \prod_{i=1}^{k} u(t - \tau_i)\, d\tau_i \tag{2.24}$$

in order to study nonlinear functionals [129]. Unlike the popular Taylor series, the Volterra series have the ability to capture memory effects and are therefore useful in order to model continuous-time dynamical systems. In fact, it has been shown that any causal time-invariant continuous nonlinear operator can be approximated by a Volterra series [20]. Also note that, when $h_k(\cdot) = 0$ for $k > 1$, we get a model in the form

$$\hat{y}(t) = \int_0^{\infty} h_1(\tau_1) u(t - \tau_1) d\tau_1,$$

where the output at time t depends linearly on past inputs, i.e. an LTI-model. Here, $h_1(t)$ is called the impulse response of the system, since it is the output we would observe with the Dirac delta function as input, i.e. $u(t) = \delta(t)$.

The discrete-time version of the Volterra series is given by [3]

$$\hat{y}(t) = \sum_{k=1}^{\infty} \sum_{\tau_1=0}^{\infty} \cdots \sum_{\tau_k=0}^{\infty} h_k(\tau_1, \ldots, \tau_k) \prod_{i=1}^{k} u(t - \tau_i).$$

System identification using Volterra series means that we determine the Volterra kernels $h_k(\tau, \ldots, \tau_k)$. However, it is in general hard to separate contributions from different kernels, making identification difficult.

Using a Gram-Schmidt orthogonalization procedure, it is possible to construct a functional series expansion using the so-called Wiener kernels [159], named after Norbert Wiener. These kernels are orthogonal when the input is a Gaussian white noise process, making it possible to separate the contribution from different kernels and thus simplifying the identification. Furthermore, if the Wiener kernels are expanded in an orthogonal Laguerre series, the resulting model can be represented by a linear dynamical block followed by a nonlinear static block [129, 20], as shown in Figure 2.1.

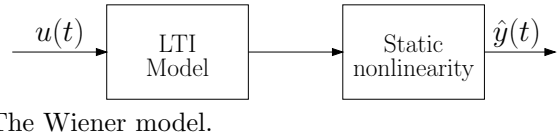


Figure 2.1: The Wiener model.

Block-oriented models

Identification using block-oriented nonlinear systems has been around for almost as long as system identification and has found an increased research activity during the last two decades. In [53], these model structures are discussed in depth.

The block-oriented nonlinear models are in a way an extension of the linear models, and they consist of dynamical linear subsystems that interact with static nonlinear elements. The goal is to identify both the linear and nonlinear blocks from the measured inputs and outputs. By combining the linear and nonlinear blocks in different ways, several popular model structures can be represented. For example:

- *Wiener models.*
  Models with the structure in Figure 2.1 are called Wiener models, because of the connection between the Wiener kernels and this structure.
- *Hammerstein models.*
  The Hammerstein model, depicted in Figure 2.2, was introduced in 1930 by A. Hammerstein, [61]. In these models, the input $u(t)$ first passes through a static nonlinearity, and then it enters the linear dynamical system, so we get kind of a mirror image of a Wiener model.

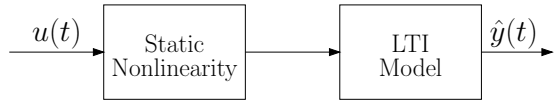


Figure 2.2: The Hammerstein model.

From the basic Hammerstein and Wiener models, more advanced nonlinear models can be built by connecting several dynamical linear and static nonlinear blocks in either series or in parallel. For example, in a Hammerstein-Wiener model, both the input and output signals are processed by static nonlinear blocks, while, in the Wiener-Hammerstein model, a static nonlinear block is sandwiched between two dynamical linear systems, see Figure 2.3.

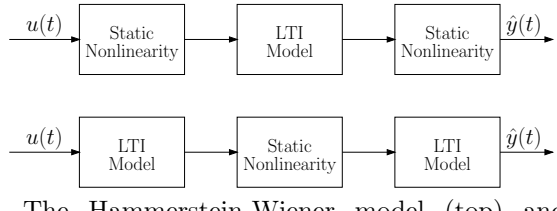


Figure 2.3: The Hammerstein-Wiener model (top) and the Wiener-Hammerstein model (bottom).

Nonlinear ARMAX models

A wide class of nonlinear models can be constructed by considering the pseudolinear regression form of the ARMAX-model in (2.20), which gives $\hat{y}(t|\theta)$ as a linear function of the regressor $\varphi(t,\theta)$. If we instead let $\hat{y}(t|\theta)$ be a nonlinear function of $\varphi(t,\theta)$, we obtain a class of models called nonlinear ARMAX (NARMAX), that can be written as

$$\hat{y}(t|\theta) = f(\varphi(t,\theta),\theta). \tag{2.25}$$

Similarly, if we let $\hat{y}(t|\theta)$ be a nonlinear function of the ARX regressor in (2.19), we get the nonlinear ARX (NARX) models

$$\hat{y}(t|\theta) = f(\varphi(t),\theta). \tag{2.26}$$

By restricting the nonlinear function $f(\cdot)$ to be within a certain family, several different model structures can be considered as subsets of the NARMAX models. For example, the block-oriented models discussed above can be considered as special cases of the NARMAX model. The same is true for many neural network architectures etc, cf. [130].

Nonlinear state-space models

As for linear models, latent variables can be used for nonlinear systems in order to represent them in state-space form

$$\begin{aligned} x(t+1) &= f(x(t),u(t)) + w(t),\\ y(t) &= g(x(t),u(t)) + v(t), \end{aligned}$$

where $x(t)$ is an auxiliary state vector, and the signals $w(t)$ and $v(t)$ are the process and measurement noise, respectively.

Function expansions

The block-oriented models, NARMAX-models, and state-space models contain a static nonlinear function $f(\cdot)$. One natural way to parameterize this function is by using a function expansion. That is, let

$$f(x) = \sum_{i=1}^{n_k} k_i f_i(x). \tag{2.27}$$

We refer to f_i as basis functions. One choice of basis functions is to use a polynomial expansion of $f(x)$. Other possible choices include the Fourier basis functions, the Laplace operator basis, wavelets, piecewise linear functions etc [88, 136, 130, 151].

Note that the parameterization in (2.27) is linear in the parameters k_i, even though the function $f(x)$ is nonlinear in the argument x. Thanks

to this property, we see that if (2.27) is used in the NARX model (2.26), then

$$\hat{y}(t|\theta) = \sum_{i=1}^{n_k} k_i f_i(\varphi(t)) = \gamma^\top(t)\theta, \tag{2.28}$$

where $\varphi(t)$ is given by (2.19) and

$$\theta = \begin{bmatrix} k_1 & \cdots & k_{n_k} \end{bmatrix}^\top,$$
$$\gamma(t) = \begin{bmatrix} f_1(\varphi(t)) & \cdots & f_{n_k}(\varphi(t)) \end{bmatrix}^\top.$$

That is, a NARX model with a given function expansion is nonlinear in the data due to the nonlinear regressor $\gamma(t)$, but it is linear in the unknown parameters θ, and it can thus be seen as a linear regression.

2.4 Estimation methods

In this section, we discuss the following problem: Given a set of identification data $\mathcal{Z}^N$, and a model structure $\mathcal{M}$, which model should we choose? From the discussion in Section 2.3.1, it seems reasonable to choose the model that minimizes the prediction errors in some sense. This strategy is discussed in Section 2.4.1. In Section 2.4.2, different statistical approaches are considered, and in Section 2.4.3 ways to incorporate the principle of parsimony in the estimation are reviewed.

2.4.1 Prediction error methods

A common family of estimation methods for system identification, that can be applied to quite arbitrary model parameterizations, are the prediction error methods (PEM) [135, 88].

The idea behind PEM is to minimize the prediction errors $\varepsilon(t,\theta) = y(t) - \hat{y}(t|\theta)$, $t = 1, \ldots, N$ when evaluated on the identification data $\mathcal{Z}^N$.

Introduce the criterion

$$V_{\text{PEM}}(\theta, \mathcal{Z}^N) \triangleq \frac{1}{N}\sum_{t=1}^{N} \ell(y(t) - \hat{y}(t|\theta)), \tag{2.29}$$

where ℓ is some scalar-valued, and usually non-negaitive, function. We then let our estimate $\hat{\theta}$ be the minimizer of (2.29), i.e.,

$$\hat{\theta} = \operatorname*{argmin}_{\theta \in \mathcal{D}_\mathcal{M}} V_{\text{PEM}}(\theta, \mathcal{Z}^N). \tag{2.30}$$

A common choice for the cost function is $\ell(\varepsilon(t,\theta)) = \|y(t) - \hat{y}(t|\theta)\|_2^2$, in which case (2.29) becomes the popular least-squares estimator.

2.4.2 Statistical approaches

As noted in Section 2.3, the prediction errors are often modeled as a stochastic process with some conditional PDF $p_\varepsilon(\varepsilon(t,\theta)|\mathcal{Z}^{t-1})$. Therefore, methods from probability theory and statistics can be applied in order to estimate a model.

The maximum likelihood method

Consider a predictor model of the form (2.3)-(2.5) that is parametrized with some vector θ. Also let

$$y \triangleq \begin{bmatrix} y^\top(1) & \cdots & y^\top(N) \end{bmatrix}^\top. \tag{2.31}$$

Using the chain rule for probability distributions and (2.6), we get

$$p(y|\theta) = \prod_{t=1}^{N} p_\varepsilon(y(t) - \hat{y}(t|\theta)|\mathcal{Z}^{t-1}, \theta). \tag{2.32}$$

By plugging the measured $y(t)$ into the above distribution and considering θ to be a free variable, we get the so-called likelihood function, which can be interpreted as the likelihood of observing the measured data for a given θ. In practice, it is often more convenient to work with the negative logarithm of the likelihood function, that is

$$V_{\mathrm{ML}}(\theta, \mathcal{Z}^N) \triangleq -\ln p(y|\theta) = -\sum_{t=1}^{N} \ln p_\varepsilon\left(y(t) - \hat{y}(t|\theta)|\mathcal{Z}^{t-1}, \theta\right). \tag{2.33}$$

The maximum likelihood (ML) estimator is then the estimator that maximizes the likelihood of the measured data, which is equivalent to minimizing the negative logarithm, i.e.

$$\hat{\theta} = \operatorname*{argmin}_{\theta \in \mathcal{D}_\mathcal{M}} V_{\mathrm{ML}}(\theta, \mathcal{Z}^N).$$

That is, out of the possible models we choose the one that assigns the highest likelihood to the data that was actually observed.

Example 2.1: Connection with PEM

To see the connection between ML and PEM, note that if $\ell(\cdot)$ in (2.29) is given by

$$\ell(\varepsilon(t,\theta)) = -\ln p_\varepsilon\left(y(t) - \hat{y}(t|\theta)|\mathcal{Z}^{t-1}, \theta\right),$$

then ML and PEM coincide. Hence, the ML-method can be seen as a way of choosing the criterion in PEM.

Also note that, if we assume that the prediction errors $\varepsilon(t,\theta)$ are independent with a Gaussian distribution of zero mean and covariance matrix Λ, then

$$-\ln p_\varepsilon(y(t) - \hat{y}(t|\theta)|\mathcal{Z}^{t-1}, \theta) = \|y(t) - \hat{y}(t|\theta)\|_\Lambda^2 + K,$$

where K is a constant that is invariant to θ. So, in this case, the maximum likelihood is the least-squares estimate. This connection was recognized by Gauss in 1809, as we discussed in Section 1.1.

Bayesian methods and MAP

In the interpretation of the ML-method above, we considered $y(t)$ to be measurements of random variables while θ is considered to be a fixed but unknown constant. In the Bayesian view, we are uncertain about the parameter vector, and thus it is considered a random variable with a prior distribution as in (2.9). However, we are certain about the data that we have measured, and we can use it to infer information about θ using Bayes' rule

$$p(\theta|\mathcal{Z}^N) = \frac{p(\mathcal{Z}^N|\theta)p(\theta)}{p(\mathcal{Z}^N)} = \frac{p(y|\theta)p(\theta)}{p(y)},$$

where the last equality follows if we consider $u(t)$ to be a known deterministic signal. The distribution $p(\theta|\mathcal{Z}^N)$ is called the posterior and describes our updated uncertainty about θ given the measured data. As seen, it is proportional to the likelihood given in (2.32) and the prior (2.9), while it is inversely proportional to

$$p(y) = \int p(y|\theta)p(\theta)d\theta,$$

where the integral is taken over the domain of θ, which we denote as $\mathcal{D}_\mathcal{M} \subseteq \mathbb{R}^{n_\theta}$. In some simple cases, these distributions can be computed analytically, but in most cases numerical methods have to be utilized.

When the posterior distribution has been computed, the system identification task can be considered done. We have, given our model, inferred all information about θ that we can from the observed data $\mathcal{Z}^N$. What to do with this distribution depends on the purpose of the system identification. If the purpose is to predict future outputs, we might for example try to compute

$$p(y(t)|\mathcal{Z}^{t-1}) = \int p(y(t)|\mathcal{Z}^{t-1},\theta)p(\theta|\mathcal{Z}^{t-1})d\theta,$$

which is a quantification of our uncertainty about $y(t)$ given all measured data until time $t-1$.

It might be inconvenient to work with distributions, especially when we cannot find an analytical expression for them, and thus we might seek a point estimate of θ. A natural choice is to use the posterior mean, i.e.

$$\hat{\theta} = \mathrm{E}_\theta\left[\theta|\mathcal{Z}^N\right].$$

However, in many cases, computing this mean might be hard, and therefore it is common to use the maximum a posteriori (MAP) estimate, i.e. the θ that maximizes $p(\theta|\mathcal{Z}^N)$. Since $p(y)$ does not depend on θ it is enough to maximize $p(y|\theta)p(\theta)$. As for the ML-method, it is here often convenient to work with the negative logarithm

$$V_{\text{MAP}}(\theta, \mathcal{Z}^N) \triangleq -\ln p(y|\theta)p(\theta) = -\ln p(y|\theta) - \ln p(\theta), \tag{2.34}$$

and notice that maximizing $p(\theta|\mathcal{Z}^N)$ is equivalent to taking the estimate as

$$\hat{\theta} = \underset{\theta \in \mathcal{D}_{\mathcal{M}}}{\operatorname{argmin}} \, V_{\text{MAP}}(\theta, \mathcal{Z}^N).$$

2.4.3 Regularization

Consider the PEM approach in Section 2.4.1 with a quadratic criterion and the linear regression model in (2.19). Then we obtain our estimate $\hat{\theta}$ by minimizing the criterion

$$V_p(\theta, \mathcal{Z}^N) = \sum_{t=1}^{N} \|y(t) - \varphi^\top(t)\theta\|_2^2. \tag{2.35}$$

Note that the dimension of θ is $n_\theta = n_a + n_b$. If the dimension is not known beforehand, we could also optimize over n_a and n_b. However, with larger n_a and n_b, we get a more flexible model, so increasing the dimension of θ is always beneficial if the only goal is to minimize the prediction errors on the data set used for identification. That is, a more complex model will always be at least as good as a less complex model if the PEM-criterion is the only consideration. This goes against the principle of parsimony discussed in Section 1.5. Also, the more complex model typically overfits the data by incorporating random disturbances present in the identification data into the model.

In order to prevent overfitting, regularization techniques can be used. This means that we add a penalty on the model complexity in some way, so that we get a criterion on the form

$$V(\theta, \mathcal{Z}^N) = V_p(\theta, \mathcal{Z}^N) + V_c(\theta), \tag{2.36}$$

where $V_p(\theta, \mathcal{Z}^N)$ is a PEM-criterion and $V_c(\theta)$ is the cost of the model complexity.

When a quadratic criterion is chosen for $V_p(\theta, \mathcal{Z}^N)$, many popular estimators can be obtained by different choices of $V_c(\theta)$, for example

- *Ridge regression:* Let the cost of complexity be chosen as $V_c(\theta) = \|\theta\|_\Lambda^2$ for some weighting matrix Λ.
- *LASSO:* The least absolute shrinkage and selection operator. Here $V_c(\theta) = \lambda\|\theta\|_1$, for some weight λ.

Both of the above choices penalize the elements in θ that deviate from 0. The LASSO-criterion also tends to give sparse solutions, in the sense that many of the elements in θ become estimated as zero [149].

The strategy of looking for sparse parameter vectors or low-rank matrices is common in system identification, since this allows us to over-parameterize the model structure and then let the estimation prune out unnecessary parameters. However, this in general results in a non-convex problem that is hard to solve. Hence, the problem of keeping the number of non-zero paramaters in a vector low is often approximated with minimization of the ℓ_1-norm, as in LASSO. The corresponding approximation for low-rank matrices is a minimization of the nuclear-norm, which has been used in e.g. Hammerstein identification [62, 46] and subspace identification [131, 155].

Example 2.2: Connection with MAP

In Example 2.1, we saw that assuming independent prediction errors $\varepsilon(t, \theta)$ with a Gaussian distribution results in a quadratic PEM-criterion when the ML-estimate is used. If we also include a Gaussian prior on θ with zero mean and covariance Λ, then

$$-\ln p(\theta) = \|\theta\|_\Lambda^2 + K,$$

where K is a constant that does not depend on θ. Hence, the MAP-criterion (2.34) will in this case be a ridge regression.

If we consider θ to have a Laplace distribution as a prior, then

$$-\ln p(\theta) = \lambda\|\theta\|_1 + K,$$

so the MAP-estimator is the same as LASSO in this case.

Hence we see how the choice of regularization can be motivated by choosing different priors on θ, and in general we can write the MAP-criterion as (2.36) by setting

$$V_p(\theta, \mathcal{Z}^N) = -\ln p(y|\theta), \quad V_c(\theta) = -\ln p(\theta).$$

2.5 Computing the estimates

In Section 2.4, different ways of choosing a model m from the model structure $\mathcal{M}$ were discussed. Most of these were based on finding the parameter vector by solving an optimization problem on the following form:

$$\hat{\theta} = \underset{\theta \in \mathcal{D}_\mathcal{M}}{\operatorname{argmin}} \; V(\theta, \mathcal{Z}^N). \tag{2.37}$$

In this section, some techniques for solving such an optimization problem are discussed, with a focus on techniques that will be utilized in this thesis.

In some cases, like for linear least-squares problems, an analytical expression for the solution to (2.37) can be found, see Section 2.5.2. However, in most cases, it is not possible to find an analytical expression, so we have to resort to numerical optimization. It is then common to use an iterative method. That is, we start with some initial guess $\hat{\theta}^{(0)}$ for the parameter vector and, in each iteration, we update it according to some rule, e.g.

$$\hat{\theta}^{(k+1)} = \hat{\theta}^{(k)} + \alpha_k f^{(k)}, \tag{2.38}$$

where $\hat{\theta}^{(k)}$ is the estimate after k iterations, $f^{(k)}$ is the search direction, and $\alpha_k > 0$ is the step length.

If $f^{(k)}$ is chosen to point in a direction in which $V(\hat{\theta}^{(k)}, \mathcal{Z}^N)$ decreases, and α_k is small enough, then we get

$$V(\hat{\theta}^{(k+1)}, \mathcal{Z}^N) \leq V(\hat{\theta}^{(k)}, \mathcal{Z}^N). \tag{2.39}$$

Hence, the value of the criterion decreases in each iteration, so we will hopefully end up at the optimal estimate. However, with only the property in (2.39), we can still end up in a local minimum of the criterion. Thus the choice of the initial estimate $\hat{\theta}^{(0)}$ will typically affect which minima we end up in, and in order to find the global minimum we have to choose the initial estimate well. In fact, (2.39) is not even enough for guaranteeing convergence to a local minim. For example, the EM algorithm, discussed in Section 2.5.5, can converge to a saddle point in certain cases [109]. However, with a suitable choice of the step-length α_k in (2.38), these methods will typically converge to a local minimum.

An important class of optimization problems is that of convex optimization. In these, there is only one minimum, so any algorithm that takes us to a local minimum will find the global minimum. There are several methods that can be used in order to find the optimal solution in these cases, see for example [21].

Finally note that an iterative scheme like (2.38) does not necessarily take into account the constraint $\theta \in \mathcal{D}_\mathcal{M}$. However, often the boundary of $\mathcal{D}_\mathcal{M}$ corresponds to the boundary of the stability region, and in this case $V(\theta, \mathcal{Z}^N)$ typically increases rapidly as θ approaches the boundary of $\mathcal{D}_\mathcal{M}$. Hence, the constraint is typically respected as long as the step-length α_k is selected in a suitable way. In other cases it might be necessary to add a step that forces the estimates to stay inside $\mathcal{D}_\mathcal{M}$, cf. Chapter 5.

2.5.1 Recursive versus batch identification

Assume that we have found $\hat{\theta}$ that minimizes $V(\theta, \mathcal{Z}^N)$. What happens if we get new measurements $y(N+1)$ and $u(N+1)$? In many optimization methods, this would mean that we have to redo the optimization from scratch, using all data in $\mathcal{Z}^{N+1}$. Such methods are here referred to as *batch methods*, or *off-line methods*.

However, if the model is already in use, it would be preferable if the previous estimate could be updated directly based on the new information without restarting the optimization from scratch. We refer to methods that work in this way as *recursive methods* or *online methods*. In a recursive method, an estimate $\hat{\theta}(t-1)$ is computed using the data available up to time $t-1$, and when the data for time t arrive an estimate $\hat{\theta}(t)$ is computed by "simple modifications" of $\hat{\theta}(t-1)$ [135]. Besides being important for online applications, such as adaptive control and fault detection, recursive algorithms typically have a running time that is linear in N, and modest memory requirements compared to other methods. Thus, a recursive algorithm may be preferable even in the offline scenario if the computational time is of concern.

2.5.2 Linear least squares

Consider a linear regression as in (2.19) or (2.28) with a quadratic criterion. Then we get an optimization problem on the form

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{t=1}^{N} \|y(t) - \varphi^\top(t)\theta\|_2^2. \tag{2.40}$$

This minimization problem can be solved analytically and, assuming that $\sum_{t=1}^{N} \varphi(t)\varphi^\top(t)$ is invertible, the solution is given by

$$\hat{\theta} = \left(\sum_{t=1}^{N} \varphi(t)\varphi^\top(t) \right)^{-1} \sum_{t=1}^{N} \varphi(t)y(t). \tag{2.41}$$

This expression can be very useful in order to analyze the resulting estimator. However, for large problems and/or small data sets, the matrix $\sum_{t=1}^{N} \varphi(t)\varphi^\top(t)$ might be ill-conditioned, so explicitly computing the inverse is often a bad idea. Luckily, since linear least squares is a very important estimator, several numerically sound methods have been developed for this problem [84].

In ridge regression, see Section 2.4.3, we add a penalty on the norm of θ and obtain the following problem:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{t=1}^{N} \|y(t) - \varphi^\top(t)\theta\|_2^2 + \|\theta\|_\Lambda^2. \tag{2.42}$$

In this case, the solution can be written as

$$\hat{\theta} = \left(\sum_{t=1}^{N} \varphi(t)\varphi^\top(t) + \Lambda \right)^{-1} \sum_{t=1}^{N} \varphi(t)y(t). \tag{2.43}$$

As can be seen, the inverse needed for ridge regression always exist when $\Lambda \succ 0$, even in cases when the inverse in (2.41) does not. Furthermore, as N increases, the effect of the regularization will decrease since Λ stays constant while $\sum_{t=1}^{N} \varphi(t)\varphi^\top(t)$ increases.

Recursive linear least squares

The solution in (2.41) can be computed in a recursive manner. In order to see this, let

$$\begin{aligned} P(t) &\triangleq \left(\sum_{k=1}^{t} \varphi(k)\varphi^\top(k) \right)^{-1} && (2.44) \\ &= \left(P^{-1}(t-1) + \varphi(t)\varphi^\top(t) \right)^{-1} && (2.45) \\ &= P(t-1) - \frac{P(t-1)\varphi(t)\varphi^\top(t)P(t-1)}{1+\varphi^\top(t)P(t-1)\varphi(t)}, && (2.46) \end{aligned}$$

where the last equality follows from the matrix inversion lemma. Note that, in the last expression, we only need a scalar inversion, instead of a full matrix inversion.

It can now be seen that

$$\begin{aligned} \hat{\theta}(t) &\triangleq P(t) \sum_{k=1}^{t} \varphi(k)y(k) && (2.47) \\ &= P(t) \left(P^{-1}(t-1)\hat{\theta}(t-1) + \varphi(t)y(t) \right) && (2.48) \\ &= \hat{\theta}(t-1) + P(t)\varphi(t) \left(y(t) - \varphi^\top(t)\hat{\theta}(t-1) \right). && (2.49) \end{aligned}$$

Thus, the estimate in (2.40) can be computed recursively as in Algorithm 1.

Algorithm 1 : Recursive linear least squares

1: $\varepsilon(t) = y(t) - \varphi^\top(t)\hat{\theta}(t-1)$.
2: $P(t) = P(t-1) - \frac{P(t-1)\varphi(t)\varphi^\top(t)P(t-1)}{1+\varphi^\top(t)P(t-1)\varphi(t)}$.
3: $\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi(t)\varepsilon(t)$.

The final question is how to initialize the algorithm. Note that the inverse in (2.44) does not exist until we have collected enough data.

Assume that the inverse in (2.44) exists for $t \geq m$. Then one option is to initialize the recursion at time $t = m$ by explicitly computing the inverse, and then we use Algorithm 1 when new data arrive. Using this initialization, our estimate $\hat{\theta}(N)$ will be the solution (2.41). Of course, using this approach, we still have to compute the matrix inverse once.

Another popular option is to let

$$P(0) = cI, \quad \hat{\theta}(0) = 0,$$

where c is some constant. In this case, we will not compute the inverse in (2.44) exactly, so our estimate of θ becomes

$$\hat{\theta}(N) = \left(\sum_{k=1}^{N} \varphi(k)\varphi^\top(k) + \frac{1}{c} I \right)^{-1} \sum_{k=1}^{N} \varphi(k) y(k),$$

which is the solution to the ridge regression problem in (2.42) with $\Lambda = (1/c)I$. However, for c large enough, the effect of the initialization is small, and the method still converges to (2.41) asymptotically as N increases.

2.5.3 Gradient-based methods

In the iterative scheme (2.38), $f^{(k)}$ is supposed to be chosen as a decent direction of $V(\hat{\theta}^{(k)}, \mathcal{Z}^N)$. A straightforward choice is to let $f^{(k)}$ be in the direction of steepest decent, which in the ℓ_2-norm is given by the negative gradient, i.e.,

$$f^{(k)} = \left[-\frac{d}{d\theta} V(\hat{\theta}^{(k)}, \mathcal{Z}^N) \right]^\top.$$

However, close to to the minimum, the steepest-decent method is often fairly inefficient, and in this case second-order information can be used as in the Newton-Raphson method where

$$f^{(k)} = \left[\frac{d^2}{d\theta^2} V(\hat{\theta}^{(k)}, \mathcal{Z}^N) \right]^{-1} \left[\frac{d}{d\theta} V(\hat{\theta}^{(k)}, \mathcal{Z}^N) \right]^\top.$$

The Gauss-Newton method

One problem of the Newton-Raphson method is that the second derivative might be cumbersome to compute, and that $f^{(k)}$ might not be a decent direction. However, if we have a quadratic criterion, then the Gauss-Newton method can be used in order to approximate the second derivative. For simplicity, consider the SISO case with the criterion given by

$$V(\theta, \mathcal{Z}^Z) = \sum_{t=1}^{N} \frac{1}{2} \varepsilon^2(t, \theta),$$

and let $\psi(t,\theta)$ be the negative gradient of $\varepsilon(t,\theta)$, i.e.

$$\psi(t,\theta) \triangleq -\left[\frac{d}{d\theta}\varepsilon(t,\theta)\right]^\top = \left[\frac{d}{d\theta}\hat{y}(t|\theta)\right]^\top. \tag{2.50}$$

It follows that

$$\frac{d}{d\theta}V(\theta,\mathcal{Z}^N) = \sum_{t=1}^{N}\psi(t,\theta)\varepsilon(t,\theta).$$

Then an approximation of the second derivative is given by

$$\frac{d^2}{d\theta^2}V(\theta,\mathcal{Z}^N) \approx R(\theta) \triangleq \frac{1}{N}\sum_{t=1}^{N}\psi(t,\theta)\psi^\top(t,\theta).$$

Using this in the Newton-Raphson method, we get the Gauss-Newton method

$$\hat{\theta}^{(k+1)} = \hat{\theta}^{(k)} + \alpha_k R^{-1}(\hat{\theta}^{(k)})\psi(t,\hat{\theta}^{(k)})\varepsilon(t,\hat{\theta}^{(k)}). \tag{2.51}$$

Note that, since $R(\theta)$ is positive definite, the Gauss-Newton method will always go in a decent direction.

Recursive Gauss-Newton

In order to make (2.51) recursive, we will assume that $\hat{y}(t|\theta)$ and $\psi(t,\theta)$ can be computed as

$$\begin{aligned}\xi(t+1,\theta) &= A(\theta)\xi(t,\theta) + B(\theta)z(t)\\ \begin{bmatrix}\hat{y}(t|\theta)\\ \mathrm{vec}(\psi(t,\theta))\end{bmatrix} &= C(\theta)\xi(t,\theta),\end{aligned}$$

where $z(t)$ is a vector containing the data measured at time t. How the matrices $A(\theta)$, $B(\theta)$ and $C(\theta)$ are computed depends on which model structure we use. This is done for several linear model structures in [91], for Wiener models in [160], and for Hammerstein models in Chapter 5 of this thesis.

Now let $\hat{\theta}(t)$ be our estimate of θ at time t. A running estimate $\xi(t)$ of $\xi(t,\theta)$ can then be computed as

$$\xi(t+1) = A(\hat{\theta}(t))\xi(t) + B(\hat{\theta}(t))z(t).$$

Note that $\xi(t)$ will not be the same as $\xi(t,\hat{\theta}(t))$, but as long as $A(\theta)$ has all eigenvalues inside the unit circle for all θ and $\hat{\theta}(t)$ converges, $\xi(t)$ will converge towards $\xi(t,\hat{\theta}(t))$. Conditions under which these assumptions hold are given in [86]. We can now define a running estimate of $\psi(t,\theta)$ and $\hat{y}(t,\theta)$ as

$$\begin{bmatrix}\hat{y}(t)\\ \mathrm{vec}(\psi(t))\end{bmatrix} = C(\hat{\theta}(t))\xi(t).$$

Then, we obtain the following running estimate of $R(\hat{\theta}(t))$

$$R(t) = \frac{1}{t}\sum_{k=1}^{t}\psi(t)\psi^{\top}(t),$$

that can be computed recursively as

$$R(t) = R(t-1) + \gamma(t)\left[\psi(t)\psi^{\top}(t) - R(t-1)\right], \tag{2.52}$$

where $\gamma(t) = 1/t$. However, as discussed in [91], it might be useful to choose $\gamma(t) > 1/t$ in order to put more weight on recent estimates. If we replace $\psi(t,\theta)$, $R(\theta)$ and $\hat{y}(t|\theta)$ with our running estimates in (2.51), then we get a recursive Gauss-Newton algorithm.

In order to ensure that our estimate $\hat{\theta}(t)$ belongs to the set $\mathcal{D}_{\mathcal{M}}$, a projection algorithm of some sort can also be added. That is, let

$$\hat{\theta}(t) = \left[\hat{\theta}(t-1) + \gamma(t)R^{-1}(t)\psi(t)\varepsilon(t)\right]_{\mathcal{D}_{\mathcal{M}}},$$

where

$$[x]_{\mathcal{D}_{\mathcal{M}}} = \begin{cases} x & \text{if } x \in \mathcal{D}_{\mathcal{M}} \\ \text{a value strictly interior to } \mathcal{D}_{\mathcal{M}} & \text{if } x \notin \mathcal{D}_{\mathcal{M}} \end{cases}.$$

The recursive Gauss-Newton algorithm can then be summarized as in Algorithm 2. Note that we, in each time-step of the algorithm, have

Algorithm 2 : Recursive Gauss-Newton

1: $\varepsilon(t) = y(t) - \hat{y}(t)$
2: $R(t) = R(t-1) + \gamma(t)\left[\psi(t)\psi(t)^{\top} - R(t-1)\right]$
3: $\hat{\theta}(t) = \left[\hat{\theta}(t-1) + \gamma(t)R^{-1}(t)\psi(t)\varepsilon(t)\right]_{\mathcal{D}_{\mathcal{M}}}$
4: $\xi(t+1) = A(\hat{\theta}(t))\xi(t) + B(\hat{\theta}(t))z(t)$
5: $\begin{bmatrix}\hat{y}(t+1)\\ \psi(t+1)\end{bmatrix} = C(\hat{\theta}(t))\xi(t+1)$

to compute the inverse of $R(t)$. In practice, it is typically better to compute $P(t) = \gamma(t)R^{-1}(t)$ recursively, which can be done by applying the matrix inversion lemma as we did in Algorithm 1 for recursive linear least squares, see for example [91] or Section 5.3.1.

2.5.4 Cyclic minimization

The parameter vector θ often has a large dimension, and might be easier to minimize with respect to one (or more) parameter at a time. This is

called cyclic minimization [143]. That is, start with an initial guess

$$\hat{\theta}^{(0)} = \begin{bmatrix} \hat{\theta}_1^{(0)} & \cdots & \hat{\theta}_{n_\theta}^{(0)} \end{bmatrix}^\top.$$

Then an improved estimate can be found by minimizing with respect to one element at a time, i.e., let

$$\hat{\theta}_i^{(k+1)} = \underset{\theta_i}{\operatorname{argmin}}\, V\left(\begin{bmatrix} \hat{\theta}_1^{(k+1)} & \cdots & \hat{\theta}_{i-1}^{(k+1)} & \theta_i & \hat{\theta}_{i+1}^{(k)} & \cdots & \hat{\theta}_{n_\theta}^{(k)} \end{bmatrix}, \mathcal{Z}^N\right)$$

for $i = 1, \ldots, n_\theta$. Then the inequality in (2.39) will hold, so the value of the criterion decreases in each iteration.

2.5.5 Majorization-minimization

The majorization-minimization approach [165] to solve (2.37) is based on finding a function that majorizes $V(\theta, \mathcal{Z}^N)$. That is, find a function $\widetilde{V}(\theta|\tilde{\theta})$ such that, for any given $\tilde{\theta} \in \mathcal{D}_\mathcal{M}$,

$$V(\theta, \mathcal{Z}^N) \leq \widetilde{V}(\theta|\tilde{\theta}), \quad \forall \theta \in \mathcal{D}_\mathcal{M} \tag{2.53}$$

with equality when $\theta = \tilde{\theta}$. Assuming that the function $\widetilde{V}(\theta|\tilde{\theta})$ is easier to minimize with respect to θ than $V(\theta, \mathcal{Z}^N)$, we let

$$\hat{\theta}^{(k+1)} = \underset{\theta \in \mathcal{D}_\mathcal{M}}{\operatorname{argmin}}\, \widetilde{V}(\theta|\hat{\theta}^{(k)}).$$

It then follows from (2.53) that

$$V(\hat{\theta}^{(k+1)}, \mathcal{Z}^N) \leq \widetilde{V}(\hat{\theta}^{(k+1)}|\hat{\theta}^{(k)}) \leq \widetilde{V}(\hat{\theta}^{(k)}|\hat{\theta}^{(k)}) = V(\hat{\theta}^{(k)}, \mathcal{Z}^N),$$

so (2.39) holds, and the value of the criterion will decrease for each iteration. In Chapter 3, we will see one example of how a majorizing function can be found.

Expectation-maximization

When the ML-criterion (2.33) is used and the model depends on a latent (or unobserved) variable z, then the expectation-maximization (EM) method [37] can be exploited in order to find a majorizing function. Note that the likelihood function can be computed as

$$p(y|\theta) = \int p(y, z|\theta) dz,$$

but the computation of this integral, let alone minimization of the resulting likelihood, might be intractable. However, if the joint log-likelihood function $p(y, z|\theta)$ is known, then we can write the criterion (2.33) as

$$V(\theta, \mathcal{Z}^N) \triangleq -\ln p(y|\theta) = \ln p(z|y, \theta) - \ln p(y, z|\theta).$$

The problem now is that z is unobserved. However, if an estimate $\tilde{\theta}$ of θ is available, then we can use the distribution $p(z|y,\tilde{\theta})$ in order to compute the expected value of the right-hand side to get

$$V(\theta, \mathcal{Z}^N) = \mathrm{E}_{z|y,\tilde{\theta}} \left[\ln p(z|y,\theta)\right] - \mathrm{E}_{z|y,\tilde{\theta}} \left[\ln p(y,z|\theta)\right].$$

So, if we let

$$Q(\theta|\tilde{\theta}) = \mathrm{E}_{z|y,\tilde{\theta}} \left[\ln p(y,z|\theta)\right], \tag{2.54}$$

then it follows that

$$V(\theta, \mathcal{Z}^N) = V(\tilde{\theta}, \mathcal{Z}^N) - Q(\theta|\tilde{\theta}) + Q(\tilde{\theta}|\tilde{\theta}) + \mathrm{E}_{z|y,\tilde{\theta}} \left[\ln \frac{p(z|y,\theta)}{p(z|y,\tilde{\theta})}\right].$$

Using Jensen's inequality it can be seen that

$$\begin{aligned}\mathrm{E}_{z|y,\tilde{\theta}} \left[\ln \frac{p(z|y,\theta)}{p(z|y,\tilde{\theta})}\right] &\leq \ln \mathrm{E}_{z|y,\tilde{\theta}} \left[\frac{p(z|y,\theta)}{p(z|y,\tilde{\theta})}\right] \\ &= \ln \int p(z|y,\tilde{\theta}) \frac{p(z|y,\theta)}{p(z|y,\tilde{\theta})} dz = 0.\end{aligned}$$

so it follows that

$$\widetilde{V}(\theta|\tilde{\theta}) \triangleq V(\tilde{\theta}, \mathcal{Z}^N) - Q(\theta|\tilde{\theta}) + Q(\tilde{\theta}|\tilde{\theta}) \geq V(\theta, \mathcal{Z}^N),$$

and hence $\widetilde{V}(\theta|\tilde{\theta})$ majorizes $V(\theta, \mathcal{Z}^N)$. Also note that minimization of $\widetilde{V}(\theta|\tilde{\theta})$ is equivalent to maximizing $Q(\theta|\tilde{\theta})$. So each iteration in the EM-method can be summarized as

- **E-step:** Evaluate the expectation

$$Q(\theta|\hat{\theta}^{(k)}) = \mathrm{E}_{z|y,\hat{\theta}^{(k)}} \left[\ln p(y,z|\theta)\right]. \tag{2.55}$$

- **M-step:** Let

$$\hat{\theta}^{(k+1)} = \underset{\theta}{\operatorname{argmax}}\ Q(\theta|\hat{\theta}^{(k)}). \tag{2.56}$$

In order to use the EM-method, we have to be able to compute the expectation in (2.55), and the method is only useful if the M-step is easier than minimizing $V(\theta, \mathcal{Z}^N)$ directly. Often, there is significant freedom in the choice of the latent variable z, and then the problem is to select it in such a way that both the E-step and the M-step become tractable. This is in general a hard problem that have inspired many research papers.

2.6 Validation

When we have found the best model in our model structure, according to some criterion, the question is if the model is good enough. If this

is not the case, then we might have to try another model structure, another criterion or even collect new data. For linear time-invariant models, there are many model validation techniques such as spectral analysis, residual analysis etc, see for example [135, 88]. Some of these methods can also be applied to nonlinear systems.

Of course, what is good enough depends, as we discussed in Section 1.4, on what our purposes are. If the goal is to construct a controller for the system, then we can design a controller based on the model and test it on the real system. If the system behaves in a satisfactory way, then the model is good enough for our purposes. In, e.g., design of a robust controller there are also formal frameworks for validating the model, see e.g. [132].

If the purpose is to use the model for prediction or simulation, it is important that the model can describe observed data, and that it generalize to data from the system that were not used during the identification. For this reason, it is common to divide the available data into at least two sets – one used for identification, and one used for validation. A first validation test is then to check if the model found using the identification data also performs well on the validation data.

2.6.1 Simulation

A common way to test the validity of a model is by simulating it using only the input. For a predictor model on the form (2.1), the simulated output $\hat{y}_s(t|\theta)$ can be generated according to

$$\begin{aligned}\hat{y}_s(t|\theta) &= g(\mathcal{Z}_s^{t-1}, \theta),\\ \mathcal{Z}_s^{t-1} &= \{(\hat{y}_s(1|\theta), u(1)), (\hat{y}_s(2|\theta), u(2)), \ldots, (\hat{y}_s(t-1|\theta), u(t-1))\}.\end{aligned}$$

Note that, in general, $\hat{y}(t|\theta) \neq \hat{y}_s(t|\theta)$ since $\mathcal{Z}_s^{t-1} \neq \mathcal{Z}^{t-1}$. A notable exception is the model structures of OE-type, since the predictor for such a model does not use the measured output, it follows that $\hat{y}(t|\theta) = \hat{y}_s(t|\theta)$, cf. (2.18) with $d_i = 0$.

Hence, when we optimize with respect to the prediction error as in PEM (see Section 2.4.1), we do not necessarily choose the model that performs best in simulations. However, in many cases, it turns out that a model that gives small prediction errors also behaves well in simulations.

2.6.2 Performance metrics

There are several performance metrics that can be used to compare two models from different model structures with each other. Here we discuss those that will be utilized in the rest of this thesis.

Since we typically view the system as a stochastic process, one option is to look at the mean squared error (MSE) of the output,

$$\text{MSE} = \frac{1}{T}\sum_{t=1}^{T} \text{E}\left[\|y(t) - \hat{y}_s(t|\hat{\theta})\|_2^2\right], \tag{2.57}$$

or the root means squared error (RMSE),

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \text{E}\left[\|y(t) - \hat{y}_s(t|\hat{\theta})\|_2^2\right]}. \tag{2.58}$$

The expectations in the above metrics can typically not be computed analytically, e.g. because the distribution of $y(t)$ is unknown. However, if we can run several trails on the real system, the expectations can be approximated using repeated simulations on different realizations of the data.

We often have only one dataset available for validation, and in such cases the FIT of the model output to the data will be used instead. The FIT is defined by

$$\text{FIT} = 100\left(1 - \frac{\|y - \hat{y}_s\|_2}{\|y - \overline{y}\mathbf{1}\|_2}\right), \tag{2.59}$$

where y is given as in (2.31) and $\hat{y}_s$ the model output $\hat{y}_s(t|\hat{\theta})$ stacked in the same way, $\overline{y}$ is the empirical mean of y and $\mathbf{1}$ is a vector of ones. Hence, FIT compares the simulated output errors with those obtained using the empirical mean as the model output. So, the FIT will be greater than zero if we perform better than just taking the output equal to the mean, and it is 100 if the simulated output is exactly equal to the true output.

Remark 2.1. *If we are primarily interested in predictions, the above metrics can be computed with the predicted output $\hat{y}(t|\hat{\theta})$ instead of the simulated output $\hat{y}_s(t|\hat{\theta})$.*

Chapter 3

Identification of nonlinear models using latent variables

3.1 Introduction

In this chapter, we consider the problem of learning a nonlinear dynamical system with multiple inputs $y(t)$ and multiple outputs $u(t)$. Generally this identification, or learning, problem can be tackled using different model structures, with the class of linear models being arguably the most well studied in engineering, statistics and econometrics [9, 16, 19, 88, 135].

Linear models are often used even when the system is known to be nonlinear [44]. However certain nonlinearities, such as saturations, cannot always be neglected. In such cases using block-oriented models is a popular approach to capture static nonlinearities [53]. To model nonlinear dynamics a common approach is to use NARMAX models [130].

In this chapter, we are interested in recursive identification methods. In cases where the model structure is linear in the parameters, recursive least-squares can be applied, see Section 2.5.2. For certain models with nonlinear parameters, the extended recursive least-squares has been used [24]. Another popular approach is the recursive prediction error method which has been developed, e.g., for Wiener models [161], polynomial state-space models [147] and Hammerstein models, see Chapter 5.

Nonparametric models are often based on weighted sums of the observed data [127]. The weights vary for each predicted output and the number of weights increases with each observed datapoint. The weights are typically obtained in a batch manner; in [7, 15] they are computed recursively but must be recomputed for each new prediction of the output.

For many nonlinear systems, however, linear models work well as an approximation. The strategy in [120] exploits this fact by first finding the best linear approximation using a frequency domain approach. Then, starting from this approximation, a nonlinear polynomial state-space model is fitted by solving a nonconvex problem. This two-step method cannot be readily implemented recursively and it requires input signals with appropriate frequency domain properties.

In this chapter, we start from a nominal predictor model structure. The nominal predictor could for instance be a linear approximation of the system but could also include known nonlinearities. The accuracy of the nominal predictor can be assessed using the prediction errors [87]. Here we characterize the nominal prediction errors using a flexible latent variable model. By jointly estimating the nominal model parameters and the statistics of the prediction errors, we can then develop a refined predictor.

The general model structure and problem formulation are introduced in Section 3.2. Then in, Section 3.3, we apply the principle of maximum likelihood to derive a statistically motivated learning criterion. In Section 3.4, this nonconvex criterion is minimized using a majorization-minimization approach that gives rise to a convex user-parameter free optimization problem. We derive a computationally efficient recursive algorithm for solving the convex problem, which can be applied to large datasets as well as online learning scenarios. The method learns parsimonious predictor models that capture nonlinear system dynamics. In Section 3.5, we evaluate the proposed method using both synthetic and real data examples.

3.2 The model structure

Let $y(t)$ denote the output of a system, and let $u(t)$ denote the inputs. Also denote all the data collected until time t as

$$\mathcal{Z}^t = \{(y(1), u(1)), (y(2), u(2)), \ldots, (y(t), u(t))\}.$$

As discussed in Section 2.3.1, the output of a system with n_y outputs can be written as

$$y(t) = y_0(t) + \varepsilon(t) \in \mathbb{R}^{n_y}, \tag{3.1}$$

where $y_0(t)$ is a predictor model, and $\varepsilon(t) = y(t) - y_0(t)$ is the prediction error. Here we use the following nominal predictor model,

$$y_0(t) = \Theta\varphi(t), \tag{3.2}$$

where the $n_\varphi \times 1$ vector $\varphi(t)$ is a given function of $\mathcal{Z}^{t-1}$ and $\Theta \in \mathbb{R}^{n_y \times n_\varphi}$ denotes the unknown parameters.

As discussed in Section 2.3.2, the popular ARX model structure, for instance, can be cast into the framework (3.1)-(3.2) by assuming that the nominal prediction error $\varepsilon(t)$ is a white noise process [88, 135]. For real-world systems this assumption is virtually never satisfied. However, as long as (3.2) accurately describes the system around its operation point, this assumption can still be a good approximation of reality. Here we are interested in cases where this may not be true, which can include systems with even weak nonlinearities [44].

Now introduce the following model of $\varepsilon(t)$ conditioned on past data $\mathcal{Z}^{t-1}$,

$$\varepsilon(t)|\mathcal{Z}^{t-1}, Z \sim \mathcal{N}(Z\gamma(t), \Sigma), \tag{3.3}$$

where $Z \in \mathbb{R}^{n_y \times n_\gamma}$ is a matrix of unknown latent variables, Σ is an unknown covariance matrix, and the $n_\gamma \times 1$ vector $\gamma(t)$ is any given function of $\mathcal{Z}^{t-1}$. This is a fairly general model structure that can capture correlated data-dependent nominal prediction errors.

Note that, when Z is zero, the prediction errors $\varepsilon(t)$ becomes a white noise process and the nominal predictor is then the optimal predictor. Hence, the purpose of Z is to model deviations from the nominal model. We model Z as a random variable, and prior to data collection we put trust in the nominal model by assuming that the prior expected value of Z is zero, and we let

$$\operatorname{vec}(Z) \sim \mathcal{N}(0, \Lambda), \tag{3.4}$$

where Λ is an unknown covariance matrix. This formulation enables us to use overparametrized error models (3.3) with a large n_γ, in which case a sparse Z is desirable.

Our goal in this chapter is to identify a refined predictor of the form

$$\hat{y}(t) = \underbrace{\widehat{\Theta}\varphi(t)}_{\hat{y}_0(t)} + \underbrace{\widehat{Z}\gamma(t)}_{\hat{\varepsilon}(t)}, \tag{3.5}$$

from a data set $\mathcal{Z}^N$. The first term is an estimate of the nominal predictor model while the second term tries to capture structure in the data that is not taken into account by the nominal model. Note that when $\widehat{Z}$ is sparse we obtain a parsimonious predictor model.

Remark 3.1. *A typical example of $\varphi(t)$ is the* ARX *regressor in* (2.19)*, i.e.,*

$$\varphi(t) = [y^\top(t-1) \ \cdots \ y^\top(t-n_a) \ u^\top(t-1) \ \cdots \ u^\top(t-n_b) \ 1]^\top, \tag{3.6}$$

in which case the nominal predictor is linear in the data and therefore captures the linear system dynamics. Nonlinearities can be incorporated if such are known about the system, in which case $\varphi(t)$ will be nonlinear in the data.

Remark 3.2. *The nonlinear function $\gamma(t)$ of $\mathcal{Z}^{t-1}$ can be understood as a basis expansion, see e.g.* (2.27)*, and can be chosen to yield a flexible model structure of the errors. In the examples below we will use the Laplace operator basis functions [136].*

Remark 3.3. *In* (3.5)*, $\hat{y}(t)$ is a one-step-ahead predictor. However, the framework can be readily applied to k-step-ahead prediction where $\varphi(t)$ and $\gamma(t)$ depend on $y(1), \ldots, y(t-k)$.*

3.3 Latent variable framework

For notational simplicity, we define a record of N samples

$$Y = \begin{bmatrix} y(1) & \cdots & y(N) \end{bmatrix} \in \mathbb{R}^{n_y \times N}.$$

In Section 3.3.1, we derive the likelihood function $p(Y|\Theta, \Lambda, \Sigma)$ and formulate a maximum likelihood estimates of the unknown parameters. Subsequently, in Section 3.3.2, we discuss the estimation of the latent variable Z, and, in Section 3.3.3, we show how all unknown parameters can be estimated jointly. The estimates $\widehat{\Theta}$ and $\widehat{Z}$ are used in (3.5). In Section 3.4, we tackle the corresponding optimization problem using a majorization-minimization approach.

3.3.1 Parameter estimation

In order to estimate the nominal model and the statistics of the error model, we need to estimate the unknown parameters $\Omega = \{\Theta, \Sigma, \Lambda\}$. We estimate Ω using the the maximum likelihood criterion discussed in Section 2.4.2. The likelihood function is given by

$$p(Y|\Omega) = \int p(Y|\Omega, Z)p(Z)dZ, \tag{3.7}$$

where the latent variable has been marginalized out.

The distribution $p(Y|\Omega, Z)$ can be computed as in (2.32), i.e.,

$$p(Y|\Omega, Z) = \prod_{t=1}^{N} p_\varepsilon(y(t) - \Theta\varphi(t)|\mathcal{Z}^{t-1}, Z), \tag{3.8}$$

where we have neglected the effect of initial conditions [135]. From (3.3) we can see that

$$p_\varepsilon(y(t) - \Theta\varphi(t)|\mathcal{Z}^{t-1}, Z) \propto \exp\left(-\frac{1}{2}\|y(t) - \Theta\varphi(t) - Z\gamma(t)\|^2_{\Sigma^{-1}}\right),$$

so it follows that

$$p(Y|\Omega, Z) = \frac{1}{\sqrt{(2\pi)^{n_y N}|\Sigma|^N}} \exp\left(-\frac{1}{2}\|Y - \Theta\Phi - Z\Gamma\|^2_{\Sigma^{-1}}\right), \quad (3.9)$$

where

$$\Phi = \begin{bmatrix}\varphi(1) & \cdots & \varphi(N)\end{bmatrix} \in \mathbb{R}^{n_\varphi \times N},$$
$$\Gamma = \begin{bmatrix}\gamma(1) & \cdots & \gamma(N)\end{bmatrix} \in \mathbb{R}^{n_\gamma \times N}.$$

The integral in (3.7) can therefore be computed using (3.9) and (3.4). In order to simplify the derivation, we introduce the vectorized variables

$$y = \mathrm{vec}(Y), \quad \theta = \mathrm{vec}(\Theta), \quad z = \mathrm{vec}(Z),$$

and

$$F = \Phi^\top \otimes I_{n_y}, \quad G = \Gamma^\top \otimes I_{n_y}.$$

Then we have the following identities

$$\mathrm{vec}(\Theta\Phi) = F\theta, \quad \text{and,} \quad \mathrm{vec}(Z\Gamma) = Gz,$$

and furthermore,

$$\|Y - \Theta\Phi - Z\Gamma\|^2_{\Sigma^{-1}} = \|y - F\theta - Gz\|^2_{I_N \otimes \Sigma^{-1}}.$$

As shown in Appendix 3.A.1, after inserting (3.9) into (3.7) we obtain

$$p(Y|\Omega) = \frac{1}{\sqrt{(2\pi)^{n_y N}|R|}} \exp\left(-\frac{1}{2}\|y - F\theta\|^2_{R^{-1}}\right), \quad (3.10)$$

where

$$R \triangleq G\Lambda G^\top + I_N \otimes \Sigma \quad (3.11)$$

is an $n_y N \times n_y N$ matrix. Note that (3.10) is not a Gaussian distribution in general, since R may be a function of Y. It can be verified that maximizing (3.10) is equivalent to solving

$$\min_{\Omega} V(\Omega), \quad (3.12)$$

where

$$V(\Omega) \triangleq \|y - F\theta\|^2_{R^{-1}} + \ln|R| \quad (3.13)$$

and $y - F\theta = \mathrm{vec}(Y - \Theta\Phi) = \mathrm{vec}\left(\begin{bmatrix}\varepsilon(1) & \cdots & \varepsilon(N)\end{bmatrix}\right)$ is nothing but the vector of nominal prediction errors. From (3.12), we can estimate the parameter matrix Θ and the covariance parameters in Λ and Σ.

3.3.2 Latent variable estimation

Next, we turn to the latent variable Z which is used to model the nominal prediction error $\varepsilon(t)$ in (3.3). The conditional distribution of Z can be written as

$$p(Z|\Omega, Y) = \frac{p(Y|\Omega, Z)p(Z)}{p(Y|\Omega)}. \tag{3.14}$$

The distributions in the right-hand side of (3.14) are given in (3.4), (3.9) and (3.10). After inserting these expressions in (3.14) we obtain, as shown in Appendix 3.A.1, a Gaussian distribution

$$p(Z|\Omega, Y) = \frac{1}{\sqrt{(2\pi)^{n_y q}|\Sigma_z|}} \exp\left(-\frac{1}{2}\|z-\zeta\|^2_{\Sigma_z^{-1}}\right), \tag{3.15}$$

with conditional mean

$$\zeta = \Lambda G^\top R^{-1}(y - F\theta). \tag{3.16}$$

and covariance matrix

$$\Sigma_z = \left(\Lambda^{-1} + G^\top (I_N \otimes \Sigma^{-1})G\right)^{-1}. \tag{3.17}$$

An estimate $\widehat{Z}$ is then given by evaluating the conditional mean of $\mathrm{vec}(Z)$ in (3.16) at the optimal estimates $\widehat{\Theta}$, $\widehat{\Sigma}$, $\widehat{\Lambda}$ obtained via (3.12).

3.3.3 Joint estimation

In summary, we obtain $\widehat{\Theta}$ and $\widehat{Z}$ via (3.12) and (3.16) for use in (3.5). We now show that these two estimates can be obtained jointly using a concise formulation (see Theorem 3.1).

Lemma 3.1. *The following equality holds*

$$\|y - F\theta\|^2_{R^{-1}} = \|y - F\theta - G\zeta\|^2_{I\otimes\Sigma^{-1}} + \|\zeta\|^2_{\Lambda^{-1}}.$$

Proof. See Appendix 3.A.1. □

Theorem 3.1. *The optimization problem*

$$\min_{\Theta, Z, \Lambda, \Sigma} \|Y - \Theta\Phi - Z\Gamma\|^2_{\Sigma^{-1}} + \|\,\mathrm{vec}(Z)\|^2_{\Lambda^{-1}} + \ln|R| \tag{3.18}$$

attains its optimal value for the Θ, Λ, Σ *that minimizes* $V(\Omega)$ *in* (3.12)*, and the minimizing* Z *is given by the conditional mean in* (3.16)*.*

Proof. First note that after vectorization

$$\|Y - \Theta\Phi - Z\Gamma\|^2_{\Sigma^{-1}} = \|y - F\theta - G\,\mathrm{vec}(Z)\|^2_{I\otimes\Sigma^{-1}}.$$

Minimizing (3.18) w.r.t Z is thus a ridge regression problem, and the solution is given by (cf. (2.43)):

$$\begin{aligned}\operatorname{vec}(\widehat{Z}) &= \left(G^\top (I_N \otimes \Sigma^{-1}) G + \Lambda^{-1}\right)^{-1} G^\top (I_N \otimes \Sigma^{-1})(y - F\theta) \\ &= \Lambda G^\top R^{-1}(y - F\theta) = \zeta.\end{aligned}$$

Concentrating out Z from (3.18) and using Lemma 3.1 yields (3.12). □

This approach to the learning of dynamical system models, based on *la*tent *va*riables, will be designated using the acronym LAVA. The non-convex problem in (3.18) can be tackled using efficient iterative search methods, e.g. based on the majorization-minimization approach.

3.4 Majorization-minimization approach

The majorization-minimization approach to solving an optimization problem is discussed in Section 2.5.5. In this approach, we first have to find a function $\widetilde{V}(\Omega|\widetilde{\Omega})$ that majorizes $V(\Omega)$ at arbitrary points $\widetilde{\Omega}$. One option is to use the EM-method (2.55)-(2.56), that is derived for the LAVA-framework in Appendix 3.B. However, there is no natural way to initialize the EM-algorithm, and in practice it typically ends up in a non-sparse local minimum.

Here we instead derive a convex majorizing function which promotes parsimonious predictor models and is amenable to a recursive formulation and also suggest a natural initial majorization point.

3.4.1 Convex majorization

In order to get a parsimonious parameterization and computationally advantageous formulation, we consider a diagonal structure of the covariance matrices in (3.3), i.e., let

$$\Sigma = \operatorname{diag}\left(\sigma_1, \sigma_2, \ldots, \sigma_{n_y}\right), \tag{3.19}$$

and let $\Lambda_i = \operatorname{diag}(\lambda_{i,1}, \cdots, \lambda_{i,q})$ denote the covariance matrix corresponding to the ith row of Z, so that

$$\Lambda = \sum_{i=1}^{n_y} \Lambda_i \otimes E_{i,i}, \tag{3.20}$$

where $E_{i,i}$ is a matrix that has a one at the ith diagonal element, and is zero everywhere else.

We begin by majorizing (3.13) with respect to the concave term $\ln|R|$:

$$\begin{aligned}\ln|R| \leq \ln|\widetilde{R}| - \operatorname{tr}\{\widetilde{R}^{-1}(I_N \otimes \widetilde{\Sigma})\} - \operatorname{tr}\{G^\top \widetilde{R}^{-1} G\widetilde{\Lambda}\} \\ + \operatorname{tr}\{\widetilde{R}^{-1}(I_N \otimes \Sigma)\} + \operatorname{tr}\{G^\top \widetilde{R}^{-1} G\Lambda\},\end{aligned} \tag{3.21}$$

where $\widetilde{\Lambda}$ and $\widetilde{\Sigma}$ are arbitrary diagonal covariance matrices and $\widetilde{R}$ is obtained by inserting $\widetilde{\Lambda}$ and $\widetilde{\Sigma}$ into (3.11). The right-hand side of the inequality above is a majorizing tangent plane to $\ln|R|$, cf. Appendix 3.A.2.

The use of (3.21) yields a convex majorizing function of $V(\Omega)$ in (3.12):

$$\begin{aligned}\widetilde{V}(\Omega|\widetilde{\Omega}) = \|y - F\theta\|^2_{R^{-1}} + \operatorname{tr}\{\widetilde{R}^{-1}(I_N \otimes \Sigma)\} \\ + \operatorname{tr}\{G^\top \widetilde{R}^{-1} G\Lambda\} + \widetilde{K},\end{aligned} \tag{3.22}$$

where $\widetilde{K} = \ln|\widetilde{R}| - \operatorname{tr}\{\widetilde{R}^{-1}(I_N \otimes \widetilde{\Sigma})\} - \operatorname{tr}\{G^\top \widetilde{R}^{-1} G\widetilde{\Lambda}\}$ only depends on the data $\mathcal{Z}^N$ and the majorization point $\widetilde{\Omega}$. Note that (3.22) is invariant with respect to $\widetilde{\Theta}$. Furthermore, it can be seen that $\widetilde{V}(\Omega|\widetilde{\Omega})$ is convex in Ω and therefore easier to minimize than the nonconvex function $V(\Omega)$.

Theorem 3.2. *The majorizing function* (3.22) *can also be written as*

$$\widetilde{V}(\Omega|\widetilde{\Omega}) = \min_Z \ \widetilde{V}(\Omega|Z, \widetilde{\Omega}) \tag{3.23}$$

where

$$\begin{aligned}\widetilde{V}(\Omega|Z, \widetilde{\Omega}) = \|Y - \Theta\Phi - Z\Gamma\|^2_{\Sigma^{-1}} + \|\operatorname{vec}(Z)\|^2_{\Lambda^{-1}} \\ + \operatorname{tr}\{\tilde{R}^{-1}(I_N \otimes \Sigma)\} + \operatorname{tr}\{G^\top \tilde{R}^{-1} G\Lambda\} + \widetilde{K}.\end{aligned} \tag{3.24}$$

The minimizing Z is given by the conditional mean in (3.16).

Proof. The problem in (3.23) has a minimizing Z which, after vectorization, equals ζ in (3.16) (see Theorem 3.1). Inserting the minimizing Z into (3.24) and using Lemma 3.1 yields (3.22). □

From Theorem 3.2, we can see that minimizing the function $\widetilde{V}(\Omega|Z, \widetilde{\Omega})$ over Ω and Z gives us the Ω that minimizes $\widetilde{V}(\Omega|\widetilde{\Omega})$ as well as the corresponding estimate of Z.

To prepare for the minimization of the function $\widetilde{V}(\Omega|Z, \widetilde{\Omega})$ in (3.24), we write the matrix qunatitates using variables that denote the ith row of the following matrices:

$$Y = \begin{bmatrix} \vdots \\ y_i^\top \\ \vdots \end{bmatrix}, \quad \Theta = \begin{bmatrix} \vdots \\ \theta_i^\top \\ \vdots \end{bmatrix}, \quad Z = \begin{bmatrix} \vdots \\ z_i^\top \\ \vdots \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \vdots \\ \overline{\gamma}_i^\top \\ \vdots \end{bmatrix}. \tag{3.25}$$

Theorem 3.3. *The minimizing arguments* Θ *and* Z *of the function* (3.24) *are obtained by solving the following convex problem:*

$$\min_{\Theta,Z} \sum_{i=1}^{n_y} \left(\|y_i - \Phi^\top \theta_i - \Gamma^\top z_i\|_2 + \|w_i \odot z_i\|_1 \right) \tag{3.26}$$

where

$$w_i = \begin{bmatrix} w_{i,1} & \cdots & w_{i,n_\gamma} \end{bmatrix}^\top \tag{3.27}$$

$$w_{i,j} = \sqrt{\frac{\overline{\gamma}_j^\top \widetilde{R}_i^{-1} \overline{\gamma}_j}{\operatorname{tr}\{\widetilde{R}_i^{-1}\}}} \tag{3.28}$$

$$\widetilde{R}_i = \Gamma \widetilde{\Lambda}_i \Gamma^\top + \widetilde{\sigma}_i I_N. \tag{3.29}$$

The minimizing covariance parameters (3.19) *and* (3.20) *are given by*

$$\hat{\sigma}_i = \frac{\|y_i - \Phi^\top \theta_i - \Gamma^\top z_i\|_2}{\sqrt{\operatorname{tr}\{\widetilde{R}_i^{-1}\}}}$$

$$\hat{\lambda}_{i,j} = \frac{|z_{i,j}|}{\sqrt{\overline{\gamma}_j^\top \widetilde{R}_i^{-1} \overline{\gamma}_j}}$$

Proof. See Appendix 3.A.3. □

Remark 3.4. *The problem in* (3.26) *is separable in the sense that each term of the sum can be optimized independently. Note that minimizing each term of* (3.26) *is equivalent to solving a* weighted square-root Lasso *problem, cf. [12, 145].*

The majorization-minimization approach proposed for solving (3.12) can thus be summarized as follows: Choose an initial majorization point Ω_0, and get an estimate of Ω by solving (3.26) with $\widetilde{\Omega} = \Omega_0$. Use this estimate as a new majorization point, and then solve (3.26) again. Keep doing this until the algorithm converges.

The next question is, at which majorization point Ω_0 should we start? Given the overparameterized error model (3.3), a natural choice is points in the parameter space which correspond to the nominal model structure. Note that, when $\Lambda = 0$, we get $Z = 0$, so we consider starting the majorization-minimization iterations from $\Omega_0 = \{\Theta_0, \Sigma_0, 0\}$.

Theorem 3.4. *Initializing the iterative majorization-minimization scheme at any point* $\Omega_0 = \{\Theta_0, \Sigma_0, 0\}$ *with* $\Sigma_0 \succ 0$ *will result in the same minimizing sequence* (Θ_n, Z_n)*, for all* $n > 0$*. Furthermore, as* $n \to \infty$*, the*

minimizing sequence (Λ_n, Σ_n) *converges to the same point irrespective of the choice of* $\Omega_0 = \{\Theta_0, \Sigma_0, 0\}$.

Proof. See Appendix 3.A.4. □

Remark 3.5. *As a result, we may initialize the majorization-minimization scheme at* $\Omega_0 = \{0, I_{n_y}, 0\}$. *This obviates the need for carefully selecting an initialization point.*

3.4.2 Recursive computation

We now show that the convex problem (3.26) can be solved recursively using cyclic minimization in each recursive step.

Computing $\widehat{\Theta}$

If we fix Z and only solve for Θ, the problem in (3.26) becomes a linear least squares problem, with the solution

$$\widehat{\Theta} = \overline{\Theta} - ZS^\top, \tag{3.30}$$

where

$$\overline{\Theta} = Y\Phi^\dagger, \quad S^\top = \Gamma\Phi^\dagger.$$

Note that both $\overline{\Theta}$ and S are idependent of Z, and that they can be computed recursively using the standard recursive least-squares algorithm described in Section 2.5.2:

$$P(t) = P(t-1) - \frac{P(t-1)\varphi(t)\varphi^\top(t)P(t-1)}{1+\varphi^\top(t)P(t-1)\varphi(t)}, \tag{3.31}$$

$$\overline{\Theta}(t) = \overline{\Theta}(t-1) + (y(t) - \overline{\Theta}(t-1)\varphi(t))\varphi^\top(t)P(t), \tag{3.32}$$

$$S(t) = S(t-1) + P(t)\varphi(t)(\gamma^\top(t) - \varphi^\top(t)S(t-1)). \tag{3.33}$$

In order to start the recursive computation, the initial values $\overline{\Theta}(0)$, $S(0)$ and $P(0)$ are needed. Natural choices for the first two are $\overline{\Theta}(0) = 0$ and $S(0) = 0$. The matrix $\Phi^\dagger$ equals $\Phi^\top\left(\Phi\Phi^\top\right)^{-1}$ when Φ has full rank, and the matrix $P(t)$ converges to $(\Phi\Phi^\top)^{-1}$. As discussed in Section 2.5.2, a common choice for the initial value of $P(t)$ is $P(t) = cI$, where a larger constant $c > 0$ leads to faster convergences of (3.31), cf. [135, 141].

Computing $\widehat{Z}$

Using (3.30), we can concentrate out Θ from (3.26) to obtain the criterion

$$\widetilde{V}(Z) = \sum_{i=1}^{n_y} \left(\|\xi_i - \left(\Gamma^\top - \Phi^\top S\right) z_i\|_2 + \|w_i \odot z_i\|_1 \right), \tag{3.34}$$

where

$$\xi_i = y_i - \Phi^\top \overline{\theta}_i.$$

In Appendix 3.C it is shown how the minimum of $\widetilde{V}(Z)$ can be found via cyclic minimization with respect to the elements of Z. This iterative procedure is implemented using recursively computed quantities and produces an estimate $\widehat{Z}(t)$ at each time t.

3.4.3 Summary of the algorithm

To solve (3.26) for a given majorization point $\widetilde{\Omega}$ recursively, we do the following for each time t:

i) Compute $\overline{\Theta}(t)$ and $S(t)$, using (3.32)-(3.31).
ii) Compute $\widehat{Z}(t)$ via the cyclic minimization of $\widetilde{V}(Z)$. Cycle through all elements L times.
iii) Compute $\widehat{\Theta}(t) = \overline{\Theta}(t) - \widehat{Z}(t)S^\top(t)$.

The estimates are initialized as $\widehat{\Theta}(0) = 0$ and $\widehat{Z}(0) = 0$. In practice, small L works well since we cycle through all elements of Z L times for each new data sample. The computational details are given in Algorithm 4 in Appendix 3.C, which can be readily implemented e.g. in MATLAB.

3.5 Numerical experiments

In this section we evaluate the proposed method and compare it with two alternative identification methods. The methods are evaluated on simulated outputs using performance metrics presented in Section 2.6.2.

3.5.1 Identification methods and experimental setup

The numerical experiments were conducted as follows. Three methods have been used: LS identification of affine ARX, NARX using wavelet networks (WAVE for short), and LAVA. From our numerical experiments, we found that performing only one iteration of the majorization-minimization algorithm produces good results, and doing so leads to a computationally efficient recursive implementation (which we denote LAVA-R for *r*ecursive), as discussed in Section 3.4.3.

For each method, the function $\varphi(t)$ is taken as the linear regressor in (3.6). Then the dimension of $\varphi(t)$ equals $p = n_y n_a + n_u n_b + 1$. For affine ARX, the model is given by

$$\hat{y}_{ARX}(t) = \overline{\Theta}\varphi(t),$$

where $\overline{\Theta}$ is estimated using recursive least squares. Note that in LAVA-R, $\overline{\Theta}$ is computed as a byproduct via (3.32).

For the wavelet network, `nlarx` in the System Identification Toolbox for Matlab was used, with the number of nonlinear units automatically detected [89].

For LAVA-R, the model is given by (3.5) and $\widehat{\Theta}, \widehat{Z}$ are found by the minimization of (3.26) using $\widetilde{\Omega}_0 = \{0, I_{n_y}, 0\}$. The minimization is performed using the recursive algorithm in Section 3.4.3 with $L = 5$. The nonlinear function $\gamma(t)$ of the data $\mathcal{Z}^{t-1}$ can be chosen to be a set of basis functions evaluated at $\varphi(t)$. Then $Z\gamma(t)$ can be seen as a truncated basis expansion of some nonlinear function. In the numerical examples, $\gamma(t)$ uses the Laplace basis expansion [136] for which the elements are given by

$$\gamma_{k_1,\ldots,k_{n_\varphi-1}}(t) = \prod_{i=1}^{p-1} \frac{1}{\sqrt{\ell_i}} \sin\left(\frac{\pi k_i(\varphi_i(t)+\ell_i)}{2\ell_i}\right), \tag{3.35}$$

where ℓ_i are the boundaries of the inputs and outputs for each channel and $k_i = 1, \ldots, M$ are the indices for each element of $\gamma(t)$. Then the dimension of

$$\gamma(t) = [\gamma_{1,\ldots,1}(t) \ \cdots \ \gamma_{n_\varphi-1,\ldots,n_\varphi-1}(t)]^\top$$

equals $n_\gamma = M^{n_\varphi-1}$, where M is a user parameter which determines the resolution of the basis.

Finally, an important part of the identification setup is the choice of input signal. For a nonlinear system it is important to excite the system both in frequency and in amplitude. For linear models a commonly used input signal is a pseudorandom binary sequence (PRBS), which is a signal that shifts between two levels in a certain fashion. One reason for using PRBS is that it has good correlation properties [135]. Hence, PRBS excites the system well in frequency, but poorly in amplitude. A remedy to the poor amplitude excitation is to multiply each interval of constant signal level with a random factor that is uniformly distributed on some interval $[0, A]$, cf. [164]. Hence, if the PRBS takes the values -1 and 1, then the resulting sequence will contain constant intervals with random amplitudes between $-A$ and A. We denote such a random amplitude sequence RS(A) where A is the maximum amplitude.

3.5.2 A system with saturation

Consider the following state-space model,

$$x_1(t+1) = \text{sat}_2[0.9x_1(t) + 0.1u_1(t)], \tag{3.36}$$
$$x_2(t+1) = 0.08x_1(t) + 0.9x_2(t) + 0.6u_2(t), \tag{3.37}$$
$$y(t) = x(t) + e(t). \tag{3.38}$$

where $x(t) = \begin{bmatrix} x_1(t) & x_2(t) \end{bmatrix}^\top$ and

$$\text{sat}_a(x) = \begin{cases} x & \text{if } |x| < a \\ \text{sign}(x)a & \text{if } |x| \geq a \end{cases}.$$

A block-diagram for the above system is shown in Figure 3.1. The measurement noise $e(t)$ was chosen as white noise with a Gaussian distribution, with covariance matrix σI where $\sigma = 2.5 \cdot 10^{-3}$.

Data were collected from the system using an RS(A) input signal for several different amplitudes A. The identification was performed using $n_a = 1$, $n_b = 1$, $M = 4$, and $N = 1000$ data samples. This means that $n_\varphi = 5$ and $n_\gamma = 256$, and therefore there are 10 parameters in Θ and 512 in Z.

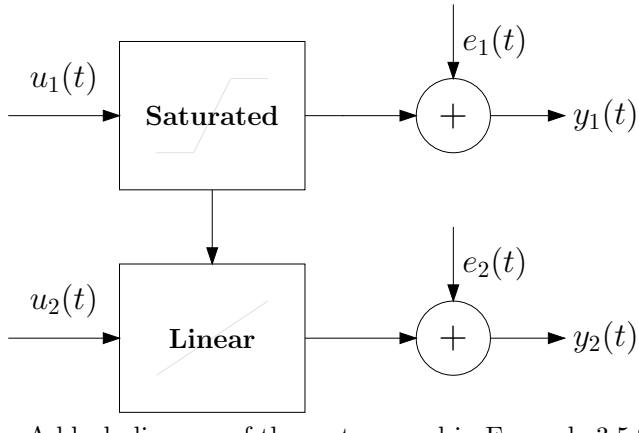


Figure 3.1: A block diagram of the system used in Example 3.5.2.

Note that, for sufficiently low amplitudes A, $x_1(t)$ will be smaller than the saturation level $a = 2$ for all t, and thus the system will behave as a linear system. However, when A increases, the saturation will affect the system output more and more. The RMSE was computed for eight different amplitudes A, and the result is shown in Figure 3.2. It can be seen that for small amplitudes, when the system is effectively linear, the ARX model gives a marginally better result than LAVA-R and WAVE. However, as the amplitude is increased, the nonlinear effects become more important, and LAVA-R outperforms both WAVE and ARX models.

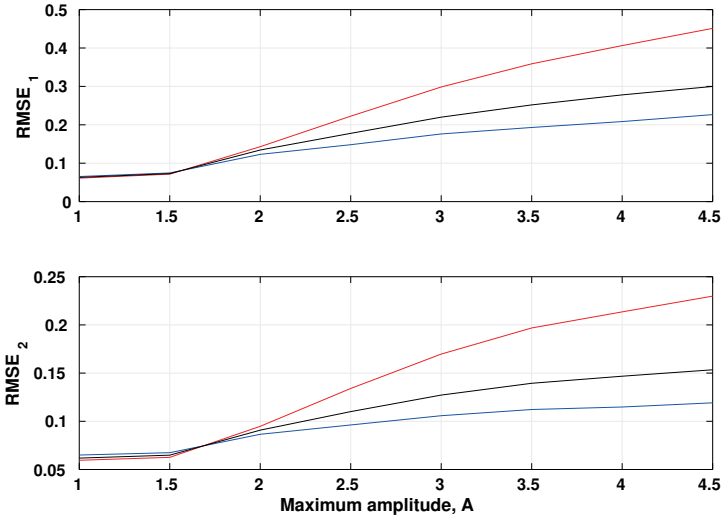


Figure 3.2: The RMSE for Example (3.5.2) computed for different input amplitudes, using LAVA-R (solid), affine ARX (dashed) and WAVE (dash-dotted).

Table 3.1: FIT for Example 3.5.3.

	LAVA-R	WAVE	ARX
Upper tank	91.6%	79.2%	84.9%
Lower tank	90.8%	76.9%	78.6%

3.5.3 A tank process

In this example a real cascade tank process is studied. It consists of two tanks mounted on top of each other, with free outlets. The top tank is fed with water by a pump. The input signal is the voltage applied to the pump, and the output consists of the water level in the two tanks. The setup is described in more detail in [164]. The data set consists of 2500 samples collected every five seconds. The first 1250 samples where used for identification, and the last 1250 samples for validation.

The identification was performed using $n_a = 2$, $n_b = 2$ and $M = 3$. With two outputs, this results in 14 parameters in Θ and 1458 parameters in Z. LAVA-R found a model with only 37 nonzero parameters in Z, and the simulated output together with the measured output are shown in Figure 3.3. The FIT values, computed on the validation data are shown in Table 3.1. It can be seen that an affine ARX model gives a good fit, but also that using LAVA-R the FIT measure can be improved significantly. In this example, WAVE did not perform very well.

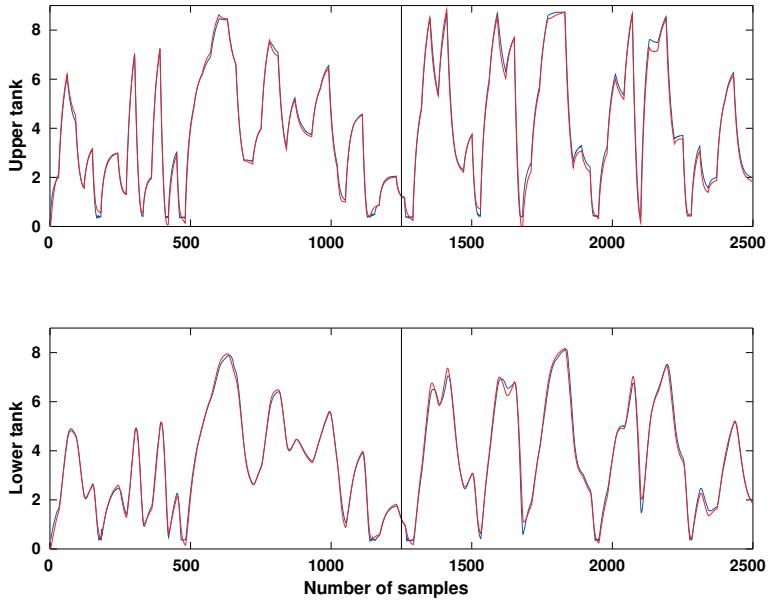


Figure 3.3: The output in Example 3.5.3 (blue), plotted together with the output of the model identified by LAVA-R (red). The system was identified using the first 1250 data samples.

3.5.4 A pick-and-place machine

In the final example, a real pick-and-place machine is studied. This machine is used to place electronic components on a circuit board, and is described in detail in [73]. The machine can be in several different modes, with two major modes being the free mode and the impact mode. In the free mode, the machine is carrying an electronic component, but is not in contact with the circuit board. When the electronic component gets in contact with the circuit board the system switches to the impact mode. Besides these two modes, the system can also exhibit saturation and other nonlinearities. The data used here are from a real physical process, and were also used in e.g. [14, 74, 118]. The data set consists of a 15s recording of the single input $u(t)$ and the vertical position of the mounting head $y(t)$. The data were sampled at 50 Hz, and the first 8s were used for identifying the model and the last 7s for validation.

The identification was performed using $n_a = 2$, $n_b = 2$ and $M = 6$. For the SISO system considered here, this results in 5 parameters in Θ and 1296 parameters in Z. LAVA-R found a model with 33 of the parameters in Z being nonzero, the output of which is shown in Figure 3.4.

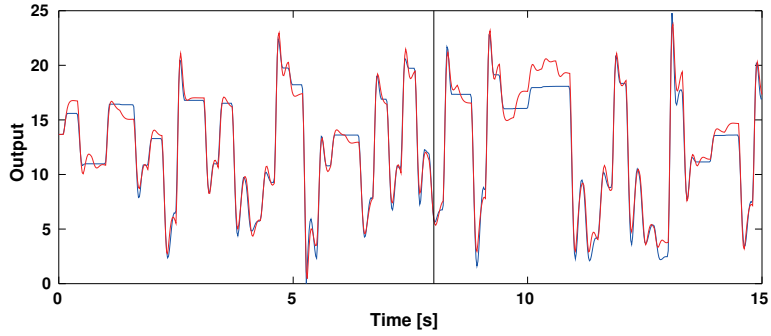


Figure 3.4: The output in Example 3.5.4 (blue), plotted together with the output of the model identified by LAVA-R (red).

Table 3.2: FIT for Example 3.5.4.

	LAVA-R	WAVE	ARX
FIT	83.2%	78.2%	73.1%

The FIT values, computed on the validation data, for LAVA-R, WAVE and affine ARX are shown in Table 3.2. LAVA-R outperforms NARX using wavelet networks, and both are better than ARX.

3.A Proofs

3.A.1 Derivation of distributions

In this appendix, the distribution in (3.7) and (3.15) are derived. In order to do so, first note the following useful equality

$$\|y - F\theta - Gz\|^2_{I_N \otimes \Sigma^{-1}} + \|z\|^2_{\Lambda^{-1}} = \|y - F\theta\|^2_{R^{-1}} + \|z - \zeta\|^2_{\Sigma_z^{-1}}, \quad (3.39)$$

where

$$\begin{aligned} R &= G\Lambda G^\top + I_N \otimes \Sigma \\ \Sigma_z &= \left(\Lambda^{-1} + G^\top (I_N \otimes \Sigma^{-1}) G\right)^{-1} \\ \zeta &= \Sigma_z G^\top (I_N \otimes \Sigma^{-1})(y - F\theta) \\ &= \Lambda G^\top R^{-1}(y - F\theta). \end{aligned}$$

This equality follows from expanding the norms on both sides of (3.39), and applying the matrix inversion lemma to see that

$$(y - F\theta)^\top R^{-1}(y - F\theta) + \zeta^\top \Sigma_z^{-1} \zeta = (y - F\theta)^\top (I_N \otimes \Sigma^{-1})(y - F\theta).$$

The sought-after distribution $p(Y|\Omega)$ is given by (3.7). By using (3.39) it follows that

$$p(Y|\Omega, Z)p(Z) \propto \exp(-\frac{1}{2}\|y - F\theta - Gz\|^2_{I_N \otimes \Sigma^{-1}}) \exp\left(-\frac{1}{2}\|z\|^2_{\Lambda^{-1}}\right) \tag{3.40}$$

$$= \exp(-\frac{1}{2}\|y - F\theta\|^2_{R^{-1}}) \exp(-\frac{1}{2}\|z - \zeta\|^2_{\Sigma_z^{-1}}), \tag{3.41}$$

with the normalization constant given by

$$\frac{1}{\sqrt{(2\pi)^{n_y(N+q)}|\Sigma|^N|\Lambda|}}.$$

Noting that

$$\int \exp\left(-\frac{1}{2}\|z - \zeta\|^2_{\Sigma_z^{-1}}\right) dZ = \sqrt{(2\pi)^{n_y q}|\Sigma_z|}$$

it follows that

$$p(Y|\Omega) = \frac{1}{\sqrt{(2\pi)^{Nn_y}|\Sigma|^N|\Lambda||\Sigma_z^{-1}|}} \exp\left(-\frac{1}{2}\|y - F\theta\|^2_{R^{-1}}\right) \tag{3.42}$$

$$= \frac{1}{\sqrt{(2\pi)^{Nn_y}|R|}} \exp\left(-\frac{1}{2}\|y - F\theta\|^2_{R^{-1}}\right), \tag{3.43}$$

which proves (3.10).

To obtain an expression for $p(Z|\Omega, Y)$, simply insert (3.41) and (3.43) into (3.14) to get

$$p(Z|\Omega, Y) = \frac{1}{\sqrt{(2\pi)^{n_y q}|\Sigma_z|}} \exp\left(-\frac{1}{2}\|z - \zeta\|^2_{\Sigma_z^{-1}}\right),$$

which is (3.15).

3.A.2 Derivation of the majorizing tangent plane

The first-order Taylor expansion of the log-determinant can be written as

$$\ln|R| \simeq \ln|\widetilde{R}| + (\partial_\sigma \ln|R|)|_{R=\widetilde{R}}(\sigma - \widetilde{\sigma}) + (\partial_\lambda \ln|R|_{R=\widetilde{R}})(\lambda - \tilde{\lambda})$$

where σ is the vector of diagonal elements in Σ and λ contains the diagonal elements in Λ.

For the derivatives with respect to λ we have

$$\frac{\partial}{\partial \lambda_{i,j}} \ln |R| = \operatorname{tr}\left(R^{-1}\frac{\partial R}{\partial \lambda_{i,j}}\right) = \operatorname{tr}\left(G^\top R^{-1} G \frac{\partial \Lambda}{\partial \lambda_{i,j}}\right).$$

Hence

$$\partial_\lambda \ln |R|_{R=\widetilde{R}}(\lambda - \tilde{\lambda}) = \operatorname{tr}\left(G^\top \widetilde{R}^{-1} G(\Lambda - \widetilde{\Lambda})\right).$$

In the same way

$$\partial_\sigma \ln |R|_{R=\widetilde{R}}(\sigma - \tilde{\sigma}) = \operatorname{tr}\left(\widetilde{R}^{-1}(I_N \otimes (\Sigma - \widetilde{\Sigma})\right).$$

Since $\ln |R|$ is concave in σ and λ, it follows that

$$\ln |R| \leq \widetilde{K} + \operatorname{tr}\left(G^\top \widetilde{R}^{-1} G\Lambda\right) + \operatorname{tr}\left(\widetilde{R}^{-1}(I_N \otimes \Sigma)\right) \tag{3.44}$$

where

$$\widetilde{K} = \ln |\widetilde{R}| - \operatorname{tr}\left(G^\top \widetilde{R}^{-1} G\widetilde{\Lambda}\right) - \operatorname{tr}\left(\widetilde{R}^{-1}(I_N \otimes \widetilde{\Sigma})\right).$$

3.A.3 Proof of Theorem 3.3

It follows from (3.20) that

$$R = \sum_{i=1}^{n_y} \left(G^\top(\Lambda_i \otimes E_{i,i})G + \sigma_i(I_N \otimes E_{i,i})\right) = \sum_{i=1}^{n_y} (R_i \otimes E_{i,i}),$$

where $R_i = \Gamma^\top \Lambda_i \Gamma + \sigma_i I_N$. It is then straightforward to show that

$$R^{-1} = \sum_{i=1}^{n_y} \left(R_i^{-1} \otimes E_{i,i}\right).$$

Hence, we can rewrite (3.24) as (to within an additive constant):

$$\widetilde{V}(\Omega|Z, \widetilde{\Omega}) = \sum_{i=1}^{n_y} \left(\frac{1}{\sigma_i}\|\bar{y}_i\|_2^2 + \|z_i\|_{\Lambda_i^{-1}}^2 + \sigma_i \operatorname{tr}(\widetilde{R}_i^{-1}) + \operatorname{tr}(\Gamma \widetilde{R}_i^{-1} \Gamma^\top \Lambda_i)\right) \tag{3.45}$$

where $\bar{y}_i = y_i - \Phi^\top \theta_i - \Gamma^\top z_i$.

We next derive analytical expressions for the Σ and Λ that minimize $\widetilde{V}(\Omega|Z, \widetilde{\Omega})$. Note that

$$\frac{\partial}{\partial \sigma_i}\widetilde{V}(\Omega|Z, \widetilde{\Omega}) = -\frac{1}{\sigma_i^2}\|\bar{y}_i\|_2^2 + \operatorname{tr}(\widetilde{R}_i^{-1}),$$

and setting the derivative to zero yields the estimate

$$\hat{\sigma}_i = \frac{\|\bar{y}_i\|_2}{\sqrt{\mathrm{tr}(\widetilde{R}_i^{-1})}}.$$

Taking the derivative with respect to $\lambda_{i,j}$,

$$\frac{\partial}{\partial \lambda_{i,j}} \widetilde{V}(\Omega|Z, \Omega_n) = -\frac{1}{\lambda_{i,j}^2} z_{i,j}^2 + \mathrm{tr}(\Gamma \widetilde{R}_i^{-1} \Gamma^\top E_{j,j}),$$

and noting that

$$\mathrm{tr}(\Gamma \widetilde{R}_i^{-1} \Gamma^\top E_{j,j}) = \overline{\gamma}_j^\top \widetilde{R}_i^{-1} \overline{\gamma}_j$$

we get the estimate

$$\hat{\lambda}_{i,j} = \frac{|z_{i,j}|}{\sqrt{\overline{\gamma}_j^\top \widetilde{R}_i^{-1} \overline{\gamma}_j}}.$$

Inserting these into (3.45), we see that we can find the minimizing Θ and Z by minimizing

$$2 \sum_{i=1}^{n_y} \left(\sqrt{\mathrm{tr}(\widetilde{R}_i^{-1})} \|y_i - \Phi^\top \theta_i - \Gamma^\top z_i\|_2 + \sum_{j=1}^{n_\gamma} |z_{i,j}| \sqrt{\overline{\gamma}_j^\top \widetilde{R}_i^{-1} \overline{\gamma}_j} \right).$$

Since term i in the above sum is invariant with respect to θ_k and z_k for $k \neq i$, we can divide term i by $2\sqrt{\mathrm{tr}(\widetilde{R}_i^{-1})}$, and see that minimizing the criterion above is equivalent to minimizing

$$\sum_{i=1}^{n_y} \left(\|y_i - \Phi^\top \theta_i - \Gamma^\top z_i\|_2 + \|w_i \odot z_i\|_1 \right).$$

3.A.4 Proof of Theorem 3.4

Initializing the majorization-minimization scheme at $\overline{\Omega}_0 = \{0, I_{n_y}, 0\}$ and $\Omega_0 = \{\Theta_0, \Sigma_0, 0\}$ where $\Sigma_0 = \mathrm{diag}\left(\sigma_1^{(0)}, \ldots, \sigma_{n_y}^{(0)}\right)$, produces two sequences denoted $\overline{\Omega}_n = \{\overline{\Theta}_n, \overline{\Sigma}_n, \overline{\Lambda}_n, \}$ and $\Omega_n = \{\Theta_n, \Sigma_n, \Lambda_n\}$ for $n > 0$, respectively. This results also in sequences $\overline{Z}_n$ and Z_n. The theorem states that:

$$\Theta_n = \overline{\Theta}_n \quad \text{and} \quad Z_n = \overline{Z}_n \tag{3.46}$$

and that

$$\Lambda_n - \overline{\Lambda}_n \to 0 \quad \text{and} \quad \Sigma_n - \overline{\Sigma}_n \to 0. \tag{3.47}$$

We now show the stronger result that the covariance matrices converge as

$$\Lambda_i^{(n)} = k_i^{(n)} \overline{\Lambda}_i^{(n)}, \quad \sigma_i^{(n)} = k_i^{(n)} \overline{\sigma}_i^{(n)}, \forall n > 0, \tag{3.48}$$

where $k_i^{(n)} = (\sigma_i^{(0)})^{\frac{1}{2^n}}$. Note that $k_i^{(n)} \to 1$ as $n \to \infty$. Hence (3.48) implies (3.47). We prove (3.48) and (3.46) by induction.

That (3.48) and (3.46) holds for $n = 1$ follows directly from Theorem 3.3. Now assume that (3.48) holds for some $n \geq 1$. Let

$$\overline{R}_i = \Gamma \overline{\Lambda}_i^{(n)} \Gamma^\top + \overline{\sigma}_i^{(n)} I_N,$$

and

$$\widetilde{R}_i = \Gamma \Lambda_i^{(n)} \Gamma^\top + \sigma_i^{(n)} I_N = k_i^{(n)} \overline{R}_i,$$

where the last equality follows by the assumption in (3.48). Therefore the weights used to estimate Θ_{n+1} and Z_{n+1} are the same as those used to estimate $\overline{\Theta}_{n+1}$ and $\overline{Z}_{n+1}$:

$$w_{i,j} = \sqrt{\frac{\overline{\gamma}_j^\top \widetilde{R}_i^{-1} \overline{\gamma}_j}{\operatorname{tr}(\widetilde{R}_i^{-1})}} = \sqrt{\frac{\overline{\gamma}_j^\top \overline{R}_i^{-1} \overline{\gamma}_j}{\operatorname{tr}(\overline{R}_i^{-1})}},$$

so we can conclude that $\Theta_{n+1} = \overline{\Theta}_{n+1}$ and $Z_{n+1} = \overline{Z}_{n+1}$. The estimate Λ_{n+1} is given by

$$\lambda_{i,j}^{(n+1)} = \frac{|z_{i,j}|}{\sqrt{\overline{\gamma}_j^\top \widetilde{R}_i^{-1} \overline{\gamma}_j}} = \frac{\sqrt{k_i^{(n)}} |z_{i,j}|}{\sqrt{\overline{\gamma}^\top \overline{R}^{-1} \overline{\gamma}}} = k_i^{(n+1)} \overline{\lambda}_{i,j}^{(n+1)}$$

so $\Lambda_i^{(n+1)} = k_i^{(n+1)} \overline{\Lambda}_i^{(n+1)}$, and in the same way it can be seen that $\sigma_i^{(n+1)} = k_i^{(n+1)} \overline{\sigma}_i^{(n+1)}$. Hence by induction (3.48) and (3.46) are true for all $n > 0$ and Theorem 3.4 follows.

3.B Derivation of LAVA-EM

In the expectation maximization method we define the function $Q(\Omega|\widetilde{\Omega})$ as in (2.54), i.e.,

$$Q(\Omega|\widetilde{\Omega}) = \mathrm{E}_{Z|Y,\widetilde{\Omega}} \left[\ln p(Y, Z|\Omega) \right].$$

3.B.1 The E-step

In the E-step (2.55) we have to compute $Q(\Omega|\widetilde{\Omega})$. Note that $p(Y, Z|\Omega) = p(Y|\Omega, Z)p(Z)$, so from (3.4) and (3.9) it follows that

$$\ln p(Y, Z|\Omega) = K - \frac{1}{2} \left(N \ln |\Sigma| + \|Y - \Theta\Phi - Z\Gamma\|_{\Sigma^{-1}}^2 + \ln |\Lambda| + \|z\|_{\Lambda^{-1}}^2 \right),$$

where K is a constant.

Denote the expected value and covariance of $\text{vec}(Z)$ given Y and $\widetilde{\Omega}$ as $\widetilde{\zeta}$ and $\widetilde{\Sigma}_z$ respectively. Then

$$\begin{aligned}\mathrm{E}_{Z|Y,\widetilde{\Omega}}\left[\|z\|^2_{\Lambda^{-1}}\right] &= \mathrm{E}_{Z|Y,\widetilde{\Omega}}\left[\mathrm{tr}\left(\Lambda^{-1}zz^\top\right)\right] \\ &= \mathrm{tr}\left(\Lambda^{-1}(\widetilde{\Sigma}_z + \widetilde{\zeta}\widetilde{\zeta}^\top)\right).\end{aligned}$$

In the same manner,

$$\begin{aligned}&\mathrm{E}_{Z|Y,\widetilde{\Omega}}\left[\|Y - \Theta\Phi - Z\Gamma\|^2_{\Sigma^{-1}}\right] = \\ &\qquad \|y - F\theta - G\widetilde{\zeta}\|^2_{I_N\otimes\Sigma^{-1}} + \mathrm{tr}((I_N\otimes\Sigma^{-1})G\widetilde{\Sigma}_z G^\top).\end{aligned}$$

Hence,

$$\begin{aligned}-2Q(\Omega|\tilde{\Omega}) &= N\ln|\Sigma| + \ln|\Lambda| + \|y - G\theta - F\widetilde{\zeta}\|^2_{I_N\otimes\Sigma^{-1}} \\ &\quad + \mathrm{tr}\left((I_N\otimes\Sigma^{-1})G\widetilde{\Sigma}_z G^\top\right) + \mathrm{tr}\left(\Lambda^{-1}(\widetilde{\Sigma}_z + \widetilde{\zeta}\widetilde{\zeta}^\top)\right) - 2K.\end{aligned}$$

3.B.2 The M-step

Note that the terms in $Q(\Omega|\widetilde{\Omega})$ that contain Λ are

$$-\frac{1}{2}\left(\ln|\Lambda| + \mathrm{tr}(\Lambda^{-1}(\widetilde{\Sigma}_z + \widetilde{\zeta}\widetilde{\zeta}^\top)\right),$$

so the optimal Λ that maximize $Q(\Omega|\widetilde{\Omega})$ is given by [142]

$$\widehat{\Lambda} = \widetilde{\Sigma}_z + \widetilde{\zeta}\widetilde{\zeta}^\top.$$

We can also see that the maximizing Θ of $Q(\Omega|\widetilde{\Omega})$ is given by

$$\begin{aligned}\widehat{\theta} &= (F^\top(I_n\otimes\Sigma^{-1})F)^{-1}F^\top(I_N\otimes\Sigma^{-1})(y - G\widetilde{\zeta}) \\ &= \left((\Phi\Phi^\top)^{-1}\Phi\otimes I\right)(y - G\widetilde{\zeta}),\end{aligned}$$

or rewritten on matrix form

$$\widehat{\Theta} = (Y - \widetilde{Z}\Gamma)\Phi^\dagger$$

where $\text{vec}(\widetilde{Z}) = \widetilde{\zeta}$, i.e., $\widetilde{Z}$ is the expected value of Z given Y and $\widetilde{\Omega}$.

If we insert $\widehat{\Theta}$ into Q, then the terms containing Σ can be written as

$$-\frac{1}{2}\left(N\ln|\Sigma| + \mathrm{tr}\left((I_N\otimes\Sigma^{-1})(G\widetilde{\Sigma}_z G^\top + \bar{y}\bar{y}^\top\right)\right),$$

where $\bar{y} = y - F\widehat{\theta} - G\widetilde{\zeta}$. As shown in [158], the above expression is minimized by

$$\widehat{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} \widetilde{X}_{ii}$$

where $\widetilde{X}_{ii}$ is the ith $n_y \times n_y$ diagonal block of $G\widetilde{\Sigma}_z G^\top + \bar{y}\bar{y}^\top$. The expectation-maximization iteration is summarized in Algorithm 3, which we denote LAVA-EM.

Note that, if we let $\widetilde{\Lambda} \to 0$, then $\widetilde{\Sigma}_z \to 0$ and thus $\widetilde{\zeta} \to 0$. In the limit, therefore, we obtain simply the estimate $\widehat{\Lambda} = 0$. Thus, if the EM algorithm is initialized at $\Omega_0 = \{\widetilde{\Theta}_0, \widetilde{\Sigma}_0, 0\}$ it will only identify the nominal predictor model.

Algorithm 3 : **Expectation maximization method for** (3.12)

1: Input: Y, Φ, Γ and $\widetilde{\Omega}$.
2: $\widetilde{\Sigma}_z = (\widetilde{\Lambda}^{-1} + G^\top(I_N \otimes \widetilde{\Sigma}^{-1})G)^{-1}$.
3: $\widetilde{\zeta} = \widetilde{\Sigma}_z G^\top (I_N \otimes \widetilde{\Sigma}^{-1})(y - F\widetilde{\theta})$.
4: $\widehat{\Lambda} = \widetilde{\Sigma}_z + \widetilde{\zeta}\widetilde{\zeta}^\top$.
5: $\widehat{\Theta} = (Y - \widetilde{Z}\Gamma)\Phi^\dagger$.
6: $\widehat{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} \widetilde{X}_{ii}$, where $\widetilde{X}_{ii}$ is the ith diagonal block of size $n_y \times n_y$ of the matrix

$$G\widetilde{\Sigma}_z G^\top + (y - F\widehat{\theta} - G\widetilde{\zeta})(y - F\widehat{\theta} - G\widetilde{\zeta})^\top.$$

7: Output: $\widehat{\Omega} = \{\widehat{\Theta}, \widehat{\Sigma}, \widehat{\Lambda}\}$.

3.C Derivation of the proposed recursive algorithm

In order to minimize $\widetilde{V}(Z)$ in (3.34) we use a cyclic algorithm. That is, we minimize with respect to one component at a time. We follow an approach similar to that in [166], with the main difference being that here we consider arbitrary nonnegative weights w_i.

Note that minimization of $\widetilde{V}(Z)$ with respect to $z_{i,j}$ is equivalent to minimizing

$$\widetilde{V}(z_{i,j}) = \|\tilde{\xi}_{i,j} - c_j z_{i,j}\|_2 + w_{i,j}|z_{i,j}|$$

where

$$\tilde{\xi}_{i,j} = \xi_i - \sum_{k \neq j} c_k z_{i,k}, \quad c_j = [\Gamma^\top - \Phi^\top S]_j.$$

As in [166] it can be shown that the sign of the optimal $\hat{z}_{i,j}$ is given by

$$\operatorname{sign}(\hat{z}_{i,j}) = \operatorname{sign}(c_j^\top \tilde{\xi}_{i,j}).$$

Hence we only have to find the absolute value $r_{i,j} = |z_{i,j}|$. Introduce the following notation:

$$\begin{aligned}\alpha_{i,j} &= \|\tilde{\xi}_{i,j}\|_2^2 \\ \beta_j &= \|c_j\|_2^2 \\ g_{i,j} &= c_j^\top \tilde{\xi}_{i,j}.\end{aligned}$$

It is then straightforward to verify that the minimization of $\widetilde{V}(z_{i,j})$ is equivalent to minimizing

$$\widetilde{V}(r_{i,j}) = (\alpha_{i,j} + \beta_j r_{i,j}^2 - 2g_{i,j}r_{i,j})^{1/2} + w_{i,j}r_{i,j},$$

over all $r_{i,j} \geq 0$, and then setting $\hat{z}_{i,j} = \text{sign}(g_{i,j})\hat{r}_{i,j}$. Note that it follows from the Cauchy-Schwartz inequality that

$$\alpha_{i,j}\beta_{i,j} \geq g_{i,j}^2.$$

Using this inequality it was shown in [166] that $\widetilde{V}(r_{i,j})$ is a convex function. The derivative of $\widetilde{V}(r_{i,j})$ is given by (dropping the subindices for now),

$$\frac{d\widetilde{V}}{dr} = \frac{\beta r - |g|}{(\beta r^2 - 2|g|r + \alpha)^{1/2}} + w. \tag{3.49}$$

Since $\widetilde{V}(r)$ is convex it follows that it is minimized by $r = 0$ if and only if $d\widetilde{V}(0)/dr \geq 0$, i.e., if and only if

$$\alpha w^2 \geq g^2. \tag{3.50}$$

Next we study the case when $g^2 > \alpha w^2$. It then follows from (3.49) that the stationary points of $\widetilde{V}(r)$ satisfy

$$(\beta r - |g|) = -w(\beta r^2 - 2|g|r + \alpha)^{1/2} \tag{3.51}$$

Solving this equation for r we get the stationary point

$$\hat{r} = \frac{|g|}{\beta} - \frac{w}{\beta\sqrt{\beta - w^2}}\sqrt{\alpha\beta - g^2}.$$

Hence we can conclude that the minimizer of $\widetilde{V}(z_{i,j})$ is given by

$$\hat{z}_{i,j} = \begin{cases} \text{sign}(g_{i,j})\hat{r}_{i,j} & \text{if } \alpha_{i,j}w_{i,j}^2 < g_{i,j}^2 \\ 0 & \text{otherwise} \end{cases}.$$

Next we show how to obtain this estimate using only recursively computed quantities. Let

$$\begin{aligned} T &= (\Gamma^\top - \Phi^\top S)^\top (\Gamma^\top - \Phi^\top S) \\ \kappa_i &= \|\xi_i\|_2^2 \\ \rho_i &= \|\xi_i - (\Gamma^\top - \Phi^\top S) z_i\|_2^2 \\ \eta_i &= \|\xi_i - (\Gamma^\top - \Phi^\top S) z_i\|_2^2 \\ \zeta_i &= (\Gamma^\top - \Phi^\top S)(\xi_i - (\Gamma^\top - \Phi^\top S) z_i) \end{aligned}$$

Then it is straightforward to show that

$$\begin{aligned} \alpha_{i,j} &= \eta_i + \beta_j z_{i,j}^2 + 2\zeta_{i,j} z_{i,j} \\ \beta_j &= T_{j,j} \\ g_{i,j} &= \zeta_{i,j} + \beta_j z_{i,j}. \end{aligned}$$

Also define $\Psi^{\mathbf{a},\mathbf{b}}(t)$ recursively, for any two vector-valued signals $\mathbf{a}(t)$, $\mathbf{b}(t)$, as

$$\Psi^{\mathbf{a},\mathbf{b}}(0) = 0 \tag{3.52}$$

$$\Psi^{\mathbf{a},\mathbf{b}}(t+1) = \Psi^{\mathbf{a},\mathbf{b}}(t) + \mathbf{a}(t)\mathbf{b}^\top(t). \tag{3.53}$$

Note that $\Psi^{\mathbf{a},\mathbf{b}}(t) = (\Psi^{\mathbf{b},\mathbf{a}}(t))^\top$. Furthermore

$$\begin{aligned} T &= \Gamma\Gamma^\top - \Gamma\Phi^\top S - S^\top \Phi\Gamma^\top + S^\top \Phi\Phi^\top S^\top \\ &= \Psi^{\gamma,\gamma}(N) - \Psi^{\gamma,\varphi}(N)S - S^\top \Psi^{\varphi,\gamma}(N) + S^\top \Psi^{\varphi,\varphi}(N)S. \end{aligned}$$

In the same way it can be verified that

$$\begin{aligned} \kappa_i &= \Psi_{i,i}^{y,y}(N) + \bar{\theta}_i^\top \Psi^{\varphi,\varphi}(N)\bar{\theta}_i - 2\bar{\theta}_i^\top [\Psi^{\varphi,y}(N)]_i \\ \rho_i &= [\Psi^{\varphi,y}(N)]_i - \Psi^{\varphi,\gamma}(N)\bar{\theta}_i \\ &\quad - S[\Psi^{\varphi,y}(N)]_i + S\Psi^{\varphi,\varphi}(N)\bar{\theta}_i \\ \eta_i &= \kappa_i - 2\rho_i^\top z_i + z_i^\top T z_i \\ \zeta_i &= \rho_i - T z_i. \end{aligned}$$

Hence, all the variables we need to find $\hat{z}_{i,j}$ can be computed using $\Psi^{\cdot,\cdot}(N)$, and the latter matrices can be computed recursively. Note that in the above derivation we did not make any assumptions on w_i, so these weights can either be precomputed, or possibly computed recursively. We can see that for the case with $\widetilde{\Omega}_0 = \{\widetilde{\Theta}_0, \widetilde{\Sigma}_0, 0\}$, the weights are given by

$$w_{i,j}^2 = \frac{1}{N}\|\overline{\gamma}_j\|_2^2 = \frac{1}{N}\left(\Gamma\Gamma^\top\right)_{j,j} = \frac{1}{N}\Psi_{j,j}^{\gamma,\gamma}(N).$$

The full algorithm for updating the needed quantities, including the update of $\overline{\Theta}$ and S, is summarized in Algorithm 4. Note that the iterations of the outer for-loop can be executed in parallel.

Algorithm 4 : **Recursive solution to** (3.26)

Input: $y(t)$, $\varphi(t)$, $\gamma(t)$, $\check{\Theta}$ and $\check{Z}$
Update $P(t)$, $\overline{\Theta}(t)$ and $S(t)$ according to (3.32)-(3.31).
Update $\Psi^{\varphi,\varphi}(t)$, $\Psi^{\gamma,\gamma}(t)$, $\Psi^{y,y}(t)$, $\Psi^{\varphi,\gamma}(t)$, $\Psi^{\varphi,y}(t)$ and $\Psi^{\gamma,y}(t)$ according to (3.53) .
$T = \Psi^{\gamma,\gamma} - \Psi^{\gamma,\varphi}S - S^\top\Psi^{\varphi,\gamma} + S^\top\Psi^{\varphi,\varphi}S$.
for $i = 1, \ldots, n_y$ **do**
    $\kappa = \Psi^{y,y}_{i,i} + \bar{\theta}_i^\top\Psi^{\varphi,\varphi}\bar{\theta}_i - 2\bar{\theta}_i^\top[\Psi^{\varphi,y}]_i$.
    $\rho = [\Psi^{\varphi,y}]_i - \Psi^{\gamma,\varphi}\bar{\theta}_i - S^\top[\Psi^{\varphi,\gamma}]_i + S^\top\Psi^{\varphi,\varphi}\bar{\theta}_i$.
    $\eta = \kappa - 2\rho^\top\check{z}_i + \check{z}_i^\top T\check{z}_i$.
    $\zeta = \rho - T\check{z}_i$
    repeat
        for $j = 1, \ldots, n_\gamma$ **do**
            $\alpha = \eta + T_{j,j}\check{z}^2_{i,j} + 2\zeta_j\check{z}_{i,j}$.
            $g = \zeta_j + T_{j,j}\check{z}_{i,j}$.
            $\beta = T_{j,j}$.
            $\hat{r} = \frac{|g|}{\beta} - \frac{w_{i,j}}{\beta\sqrt{\beta - w^2_{i,j}}}\sqrt{\alpha\beta - g^2}$.
            $\hat{z}_{i,j} = \begin{cases} \text{sign}(g)\hat{r} & \text{if } \alpha w^2_{i,j} < g^2 \\ 0 & \text{otherwise} \end{cases}$
            $\eta := \eta + T_{j,j}(\check{z}_{i,j} - \hat{z}_{i,j}) + 2(\check{z}_{i,j} - \hat{z}_{i,j})\zeta_j$.
            $\zeta := \zeta + [T]_j(\check{z}_{i,j} - \hat{z}_{i,j})$.
            $\check{z}_{i,j} := \hat{z}_{i,j}$.
        end for
    until number of iterations equals L.
end for
$\widehat{Z} = \check{Z}$.
$\widehat{\Theta} = \overline{\Theta} - \widehat{Z}S$.
Output: $\widehat{\Theta}, \widehat{Z}$.

Chapter 4

Identification of PWARX models

4.1 Introduction

In this chapter, we consider the problem of identifying a nonlinear model from a MIMO dynamical system using a finite record of its output $y(t)$ and input $u(t)$. As discussed in Section 2.3.1, a broad range of nonlinear dynamical systems can be modeled as

$$\hat{y}(t) = g(\varphi(t)), \tag{4.1}$$

where $\varphi(t)$ is a function of past inputs and outputs, and $g(\cdot)$ is the unknown function of interest. By choosing the regression vector $\varphi(t)$ as

$$\begin{aligned} \varphi(t) = [-y^\top(t-1) \quad \cdots \quad -y^\top(t-n_a) \\ u^\top(t-1) \quad \cdots \quad u^\top(t-n_b) \quad 1]^\top \end{aligned} \tag{4.2}$$

we get an NARX-model. In this chapter, we focus on the NARX model structure, but the proposed method can be applied to any regressors $\varphi(t)$ that can be computed using data collected until time $t-1$.

The identification problem is then to find the unknown function $g(\cdot)$. If $g(\cdot)$ was allowed to be any function, the identification problem would be very challenging. In Section 2.3.3, we saw how we can get different model structures by restricting $g(\cdot)$ to some class of functions. In this chapter, we consider the class of piecewise affine functions. These functions are known for their universal approximation properties [85, 22], and are therefore popular in system identification. Using a piecewise affine function in (4.1)-(4.2) gives the flexible class of *piecewise* ARX models (PWARX). In such models, the space in which $\varphi(t)$ resides is partitioned into separate regions and a local linearization of $g(\cdot)$ is used for each region. The resulting models can approximate a general nonlinear model, and are also useful for systems that change their modes,

e.g., due to saturations, and they have been shown to be useful for both prediction and control of nonlinear systems [137, 13].

Even if the identification of (4.1) becomes simpler when we restrict $g(\cdot)$ to be a piecewise affine function, this is still a complex task. In fact, it was shown in [83] that identifying $g(\cdot)$ using the prediction error method in Section 2.4.1 is an NP-hard problem in general. The main problem here is that the regions and the model parameters in each region have to be estimated simultaneously. Extensive surveys of recent methods and results can be found in [121] and [50].

In [126], it was shown how the PWARX identification problem, under relatively mild conditions, can be reformulated as a mixed-integer linear or quadratic program. However, even though heuristic algorithms for solving such a program exist, the associated problem is still NP-hard, and can be prohibitively time consuming for larger data sets. In [156], an algebraic approach was developed for the noise-free case, but the resulting method is rather sensitive to noise compared to other approaches. To deal with noisy data, a bounded-error approach was proposed in [14], which decides the number of regions by a user-specified bound on the prediction error. This results in an NP-hard optimization problem, which is solved using a greedy algorithm that finds a suboptimal solution.

If the regions are given, the problem is reduced to finding the linear submodels for each region. Different heuristics for an initial clustering of the data into regions have been suggested in the literature, such as the k-means like method in [48], and an expectation-maximization method in [115]. In [72], a Bayesian approach was used that alternates between updating the submodels and assigning new samples to each cluster in a greedy manner. A similar approach was taken in [8], where a recursive method was developed in which each new sample is assigned to a region and then the corresponding parameters are locally optimized. Common to these methods is that they find suboptimal solutions, and that good initializations and choice of user parameters are typically needed in order to get a good estimate.

The approach proposed in [118] estimates a linear submodel for each observation and penalizes the number of unique submodels. Such a penalization leads to a regularized nonconvex optimization problem, but a convex relaxation using a weighted sum of norms was proposed to tackle it. The unique solution to the relaxed problem is, however, highly dependent on carefully selecting the regularization parameters. Furthermore, as the number of model parameters increases with the number of data points, the computational requirements become prohibitive for large data records.

The method proposed in this chapter is based on selecting a set of linearization points of the nonlinear system (4.1) and then using the

LAVA-framework of Chapter 3 in order to identify the unknown parameters. As a result, we get a statistically motivated, and tuning-parameter free, convex optimization problem that can be solved recursively. These features address important limitations of the aforementioned existing methods. The set of linearization points, which form the regions of the local linear models, are selected using a data-adaptive clustering technique. This is similar to the approach in [48] and [115]. However, while those methods are constructed for cases with few clusters, the use of the LAVA-framework of Chapter 3 makes it possible to automatically penalize and prune out regions with similar linear dynamics – thus allowing the user to initially overparameterize the model. At the same time, this approach eliminates the need for carefully tuned user parameters as in e.g. [118]. The proposed method automatically identifies a predictive PWARX model of the nonlinear system after selecting the model order and the number of linearization points. Furthermore, the resulting convex problem is solved with a complexity that grows linearly with the number of data points and the solution method is therefore well-equipped to tackle large datasets.

The chapter is organized as follows. The model and problem formulation are presented in Section 4.2, followed by a discussion about selecting the linearization points in Section 4.3. The proposed identification method is presented in Section 4.4 together with a discussion about different regularization techniques. A summary of the proposed method is presented in Section 4.5. Finally, in Section 4.6, the proposed method is tested on both simulated and real data sets.

4.2 The PWARX model

A well studied subclass of the NARX models is the affine ARX models. As discussed in Section 2.3.2, these are models where $\hat{y}(t)$ depends linearly on the $n_\varphi \times 1$ vector $\varphi(t)$, i.e.,

$$\hat{y}(t) = \Theta\varphi(t) \tag{4.3}$$

where $\Theta \in \mathbb{R}^{n_y \times n_\varphi}$ denotes the matrix of unknown parameters. These models can be used to approximate any linear system [92]. Affine system models as in (4.3) are also useful as local approximations of nonlinear systems, but they cannot capture nonlinear dynamics.

In [8], it was shown how a general NARX model (4.1) can be approximated by linearizing $g(\varphi)$ around a set of points $\mu_1, \ldots, \mu_{n_r} \in \mathbb{R}^{n_\varphi}$, and then let the linearized submodel around μ_i be used in the region

$$\mathcal{R}_i = \{\varphi \in \mathbb{R}^{n_\varphi} \mid \|\varphi - \mu_i\|_2 \leq \|\varphi - \mu_j\|_2, \forall j\}, \tag{4.4}$$

which is a convex polyhedron. That is, we let

$$\hat{y}(t) = \Theta_i \varphi(t), \quad \text{if } \varphi(t) \in \mathcal{R}_i, \tag{4.5}$$

where the parameter matrix is now allowed to depend on the region of the regressor space $\mathbb{R}^{n_\varphi}$ to which $\varphi(t)$ belongs. This is a nonlinear model that is piecewise affine in the regressor space, and this type of models are thus called piecewise affine ARX (PWARX) models. In general, the regions in a PWARX model can be chosen as any convex polyhedron, and when we let $n_r = 1$ we get the affine ARX model.

Even though (4.5) is a nonlinear model, it can be formulated as a linear regression if the regions $\mathcal{R}_i$ are given. This is done by stacking all parameters into one matrix, i.e.,

$$\hat{y}(t) = \vartheta \phi(t) \tag{4.6}$$

where

$$\vartheta = \begin{bmatrix} \cdots & \Theta_i & \cdots \end{bmatrix} \in \mathbb{R}^{n_y \times n_r n_\varphi}, \; \phi(t) = \begin{bmatrix} \vdots \\ f_i\left(\varphi(t)\right) \\ \vdots \end{bmatrix} \in \mathbb{R}^{n_r n_\varphi} \tag{4.7}$$

and f_i is an indicator function

$$f_i\left(\varphi(t)\right) = \begin{cases} \varphi(t) & \text{if } \varphi(t) \in \mathcal{R}_i \\ 0 & \text{otherwise} \end{cases}. \tag{4.8}$$

Assuming that the regions $\mathcal{R}_i$ are given, an estimate of ϑ can be found via linear least squares, as in Section 2.5.2. However, when we use the PWARX model to identify a general nonlinear system, the regions are typically not given beforehand, and we somehow have to choose both the number of linearization points n_r, and their locations μ_i.

To tackle this problem, one approach is to overparametrize the model in (4.5) by choosing n_r to be large and thus yielding a fine partitioning of the regressor space. This approach is pursued in this chapter, and in Section 4.3 methods for selecting linearization points are discussed.

In an overparamterized model, the standard least-squares method is inadequate without some kind of regularization. In Section 4.4, different regularization approaches are discussed, and a recursive user parameter-free method, based on the LAVA-framework of Chapter 3, is proposed.

Remark 4.1. *In this chapter, we focus on the* PWARX *model structure. However, the method proposed in Section 4.5 is not restricted to regressors of the form* (4.2)*. In fact, any regressor* $\varphi(t)$ *that can be computed from* $\mathcal{Z}^{t-1}$ *can be used.*

4.3 Selection of the linearization points

In general, there is no prior information about how many linearization points that are needed in order to get a useful approximation of the true nonlinear system. As mentioned in Section 4.2, the approach taken here is to choose n_r "large", and thus effectively overparametrizing the problem.

The next problem is to decide around which points the model should be linearized, i.e., where to place μ_i. This can be done in several ways. One approach is to let each observed regression vector $\varphi(t)$ be a linearization point, thus creating one region for each observation (hence $n_r = N$), cf. [118]. In this case, the number of parameters $n_r n_y n_\varphi$ to estimate will grow linearly with N, which renders the identification problem intractable for large datasets.

For fixed n_r, an alternative approach is to arrange the linearization points μ_i in an uniform lattice that covers the parts of the regressor space that we are interested in, thus giving n_r rectangular linearization regions. However, such a partitioning does not take into account that some parts of the regressor space will contain more data than others, and hence they will be more informative.

A common approach to cluster data is to use k-means clustering [16]. In this approach, the observed regression vectors $\varphi(t)$ are clustered into n_r sets $\{S_1, \ldots, S_{n_r}\}$, that are obtained by solving the following problem:

$$\min_{S_1,\ldots,S_{n_r}} \sum_{i=1}^{n_r} \sum_{\varphi(t)\in S_i} \|\varphi(t) - \mu_i\|_2^2,$$

where the linearization points $\{\mu_i\}$ are the means of each set of discrete points S_i, i.e.,

$$\mu_i = \frac{1}{|S_i|} \sum_{\varphi(t)\in S_i} \varphi(t).$$

This is an NP-hard problem, but efficient heuristic algorithms exist which scale well with the dataset size [93, 119, 5]. The regions are then determined by the resulting linearization points together with (4.4). Using k-means clustering leads to a data-adaptive partioning of the regressor space, and tends to give solutions with a finer partitioning in parts of the regressor space where we have more measured data.

Alternatives to the k-means approach include variants of hierarchical clustering and the k-harmonic means method [49, 60]. Hierarchical clustering provides clusters with varying granularity, but is more computationally complex than the k-means approach.

Remark 4.2. *If the observed data were generated by a* PWARX *model, with the number of regions being known, the regions found by e.g. k-*

means usually would not be the same as the true regions. For this reason it is desirable to use a fine partitioning, i.e. choose n_r to be significantly larger than the true number of regions.

Remark 4.3. *Using rectangular linearization regions $\mathcal{R}_i$ simplifies the implementation of the identification methods discussed below. When k-means is used, this can be achieved by performing the clustering in each dimension of $\mathbb{R}^{n_\varphi}$ separately.*

4.4 Identification method

In Section 2.4, different parameter estimators are discussed. As mentioned, a natural choice is to minimize the prediction errors in some sense. Since we assume that the model is heavily overparameterized, the standard linear least-squares estimator is not an adequate choice and some kind of regularization is needed.

Some common regularization techniques are discussed in Section 2.4.3. However, there are ways to tailor the regularization to the problem we consider in this chapter.

4.4.1 Sum-of-norm regularization

In [118], a criterion for identifying single-input/single-output PWARX-models was suggested. The idea is to exploit the fact that, in a finely partitioned regressor space, neighboring regions $\mathcal{R}_i$ are likely to exhibit similar dynamics. In [118] the regions are chosen in such a way that there is only one observed regressor $\varphi(t)$ in each region $\mathcal{R}_i$. Hence, it is reasonable to assume that for many pairs of regions, the corresponding parameter vectors should be nearly the same, i.e., $\|\Theta_i - \Theta_j\|_2$ is close to zero. The method proposed in [118] makes use of a sum-of-norms regularization of the least-squares method (SNR-LS), that penalizes the weighted ℓ_2-norm of all pairwise differences $\|\Theta_i - \Theta_j\|_2$, and can be written as

$$\min_{\Theta_t} \sum_{t=1}^{N} \left(y(t) - \Theta_t \varphi(t)\right)^2 + \lambda \sum_{t=1}^{N} \sum_{s=1}^{N} K(t,s) \|\Theta_s - \Theta_t\|_2, \tag{4.9}$$

where $K(\cdot,\cdot)$ is some user-defined kernel and $\lambda > 0$ is a weight the user has to tune. In order to take into account that nearby points in the regressor space are more likely to be from the same region, the authors

of [118] suggested using the kernel,

$$K_\ell(i,j) = \begin{cases} 1, & \text{if } \varphi(i) \text{ is one of the } \ell \text{ closest neighbors of} \\ & \varphi(j) \text{ among all observations,} \\ 0, & \text{otherwise.} \end{cases} \tag{4.10}$$

when identifying a PWARX-model. Note that the number of unknown parameters in this method is $n_\varphi N$, and thus it will be computationally intractable for large datasets. Furthermore, when the parameters Θ_t, $t = 1, \ldots, N$, have been identified, we still have to estimate the shape of each region. The suggestion in [118] is to use a classification algorithm. As an example, a support vector machine is utilized, but the authors note that such a classification algorithm may perform poorly on more complicated regions.

In order to find a more efficient way of identifying a PWARX-model, we next develop a method based on the LAVA-framework.

4.4.2 Proposed method

The idea in the LAVA-framework, described in Chapter 3, is to estimate a nominal predictor model from a restricted model structure, and simultaneously estimate the statistics of the prediction errors from a very flexible model structure. The nominal predictor can then be refined based on the estimated error statistics.

In order to see how this way of thinking can be applied to PWARX-models, we first reparameterize the model in (4.5). Assume that we have chosen n_r regions as in Section 4.3, and consider the model in region $\mathcal{R}_1$ to be our nominal model with parameters given by $\Theta_1 = \Theta$. Note that any ordering of the regions is possible, so there is no loss of generality in assuming that the nominal model is the one in region $\mathcal{R}_1$.

As discussed in Section 4.4.1, it is reasonable to assume that neighboring regions in the overparameterized model have similar dynamics. Hence, for the remaining regions, we let the parameters be formed by a set of differences $\delta_j \in \mathbb{R}^{n_y \times n_\varphi}$, $j = 1, \ldots, n_d$, from Θ. That is, in region $\mathcal{R}_i$, we let the parameters be given by

$$\Theta_i = \Theta + ZD_i,$$

where

$$Z = \begin{bmatrix} \delta_1 & \cdots & \delta_{n_d} \end{bmatrix} \in \mathbb{R}^{n_y \times n_\varphi n_d},$$

D_i is some linear transformation of the differences, and $D_1 = 0$. An example of such a reparameterization is given in Example 4.1.

The output of our predictor model will then be

$$\hat{y}(t) = (\Theta + ZD_i)\,\varphi(t), \quad \text{if } \varphi(t) \in \mathcal{R}_i. \tag{4.11}$$

Similarly to (4.6)-(4.8), we can rewrite this as

$$\hat{y}(t) = \Theta\varphi(t) + Z\gamma(t), \tag{4.12}$$

where

$$\gamma(t) = D \begin{bmatrix} f_2(\varphi(t)) \\ \vdots \\ f_{n_r}(\varphi(t)) \end{bmatrix} \in \mathbb{R}^{n_\varphi n_d}, \tag{4.13}$$
$$D = \begin{bmatrix} D_2 & \cdots & D_{n_r} \end{bmatrix} \in \mathbb{R}^{n_\varphi n_d \times n_\varphi (n_r - 1)},$$

and $f_i(\varphi)$ is the indicator function given in (4.8).

Example 4.1: Example of differences

To get an intuitive feeling for the parameterization used here, we give two different examples.

A simple parameterization is to let the parameters in $\mathcal{R}_i$ be given by $\Theta_i = \Theta + \delta_i$, $i > 1$. In this case the difference matrix becomes $D = I_{n_\varphi(n_r-1)}$. As long as most regions have parameter matrices that share elements with the nominal matrix Θ, the resulting Z will be sparse.

If we instead assume that the parameters in $\mathcal{R}_i$ should be close to the parameters in $\mathcal{R}_{i-1}$, the following parameterization might give a sparser Z

$$\begin{aligned} \Theta_1 &= \Theta \\ \Theta_2 &= \Theta_1 + \delta_1 = \Theta + \delta_1 \\ \Theta_3 &= \Theta_2 + \delta_2 = \Theta + \delta_1 + \delta_2 \\ &\vdots \\ \Theta_{n_r} &= \Theta_{n_r-1} + \delta_{n_r-1} = \Theta + \delta_1 + \delta_2 + \cdots + \delta_{n_r-1}. \end{aligned}$$

That is, the parameters for $\mathcal{R}_i$, $i > 1$, are expressed as a cumulative sum of differences from the parameters in $\mathcal{R}_1$.

For this example $n_d = n_r - 1$, and the incremental difference matrix becomes

$$D = \begin{bmatrix} I & I & \cdots & I \\ & I & \cdots & I \\ & 0 & \ddots & \vdots \\ & & & I \end{bmatrix}.$$

In Section 4.5.1 a different reparametrization is suggested.

To interpret this in the framework of Chapter 3, let $y_o(t) = \Theta\varphi(t)$ be the nominal predictor. Then the nominal prediction error is given by $\varepsilon(t) = y(t) - \Theta\varphi(t)$, i.e., the prediction error we would get if the nominal (linear) predictor was used in all regions. It then follows that the mean of the prediction error should depend on which region we are in, and we assume that

$$\varepsilon(t)|\mathcal{Z}^{t-1}, Z \sim \mathcal{N}(Z\gamma(t), \Sigma).$$

so that the mean, conditioned on $\mathcal{Z}^{t-1}$ and Z, is equal to $ZD_i\varphi(t)$ if $\varphi(t) \in \mathcal{R}_i$. Since Z describes the differences in dynamics between neighboring regions, we assume prior to data collection that

$$\text{vec}(Z) \sim \mathcal{N}(0, \Lambda). \tag{4.14}$$

The majorization-minimization scheme in Section 3.4 can now be used to estimate Θ, Z, Σ and Λ. As suggested there, we use $\widetilde{\Omega} = \{0, I_{n_y}, 0\}$ as our majorizing point, and we only consider one iteration in the MM-scheme. Using the notation in Chapter 3, see e.g. (3.25), and Theorem 3.3, the approach thus reduces to solving the following convex optimization problem:

$$\min_{\Theta,Z} \sum_{i=1}^{n_y} \left(\|y_i - \Phi^\top \theta_i - \Gamma^\top z_i\|_2 + \|w \odot z_i\|_1 \right), \tag{4.15}$$

where

$$w = \begin{bmatrix} w_1 & \cdots & w_{n_\varphi n_d} \end{bmatrix},$$
$$w_i = \frac{\|\bar{\gamma}_i\|_2}{\sqrt{N}}.$$

Another way to view the above discussion is that choosing $\gamma(t)$ according to (4.13) in the model structure of Chapter 3 yields a refined predictor that is a PWARX-model. Furthermore, since the estimates of the latent variable Z tend to be sparse, the method tends to combine regions with similar dynamics by estimating the same refined parameter matrix in several regions.

4.5 Summary of the proposed method

For PWARX-models, the resulting *r*ecursive *i*dentification method using *l*ocally *l*inearized models (RILL) can be summarized as follows:
1) Set the model orders n_a and n_b and choose n_r.
2) Choose the n_r linearization points using $\{\varphi(t)\}_{t=1}^N$.

3) Construct incremental difference matrix D.
4) Solve (4.15) recursively, using Algorithm 4 in Appendix 3.C.
In Step 1, the integer parameters of the model have to be decided. A practical method for this is to perform cross-validation on the data set $\{u(t), y(t)\}_{t=1}^{N}$. That is, use the first N' samples to identify the model for a given triplet. Then, predict the output of the remaining $N - N'$ samples via (4.12) and choose the triplet which yields the minimum sum of squared output errors.

In Step 2-3, the linearization points and the difference matrix D have to be chosen. This can be done in many ways, but in Section 4.5.1 we discuss a standard way of doing this that reduces the number of choices that has to be made by the user.

Finally, in Step 4, the problem in (4.15) is solved recursively. Using Theorem 3.3, the likelihood function $p(Y|\Omega)$ can also be further minimized by iterating the majorization-minimization scheme.

4.5.1 Incremental differences

Here we consider regions as in (4.4), and thus Step 2 in the proposed method amounts to choosing a set of linearization points μ_i. In the case there is no prior knowledge to use in the selection of linearization points, we propose to place them in a lattice using k-means clustering in each dimension separately. This yields rectangular linearization regions, and as we will see this simplifies the selection of the difference matrix D. Also, using k-means ensures that the data are well spread across the regions.

In Step 3, the difference matrix D has to be determined. The parameterization we propose to use is illustrated in Figure 4.1 for the case when the linearization regions are divided only with respect to two variables in $\varphi(t)$, e.g. $u(t-1)$ and $y(t-1)$ in the scalar case. That is, let the middle region correspond to Θ, and let the incremental differences δ_i extend either vertically or horizontally. This generalizes into higher dimensions if needed. Thus, the only choice the user has to make is the size of the grid and in which direction the differences should extend, where the latter choice is binary.

4.6 Numerical evaluation

In this section, RILL is evaluated on several examples, both using simulated data and real data. As a performance metric we will use FIT defined in Section 2.6.2 and the normalized MSE (NMSE). The NMSE is

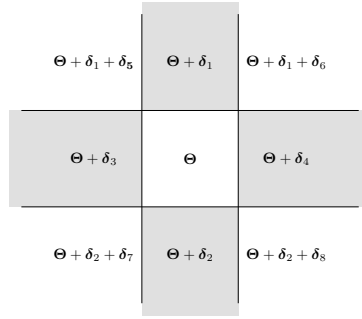


Figure 4.1: Stylized example of the linearization regions and parameterization used, where $n_r = 9$ (3×3 grid). The middle region is chosen as the reference $\mathcal{R}_1$ with the nominal model, and the differences δ_j as illustrated.

defined by

$$\text{NMSE} = \frac{\text{MSE}}{\text{MSE}_o}$$

where MSE is defined in Section 2.6.2, and MSE_o is the mean square error corresponding to the true parameter vector θ_o and the true regions $\mathcal{R}_i^o$, cf. [140],

$$\text{MSE}_o = \frac{1}{T}\sum_{t=1}^{T} \text{E}\left[\|y(t) - \hat{y}_s(t|\theta_o)\|_2^2\right].$$

Of course, the Nmse as defined above can only be computed when we generate data from a Pwarx-system where we know the true regions and parameters, so Fit will be used when this is not the case.

4.6.1 Setup of identification methods

In this section, we will describe how the numerical experiments were conducted. Three methods have been used: Rill, Snr-LS [118] and the affine Arx model in (4.3).

For Rill we follow the steps in Section 4.5. In particular, we have chosen the linearization points and incremental differences as described in Section 4.5.1. For all examples we have used a 9×9 grid of linearization points. Therefore only the model orders n_a, n_b and the vertical/horizontal orientation of the differences have to be chosen.

The Snr-Ls method in [118] uses a sum of-norm-regularization as in (4.9), and here we use the kernel (4.10). In this method, the user has to specify the regularization parameter λ and ℓ. In each example, we have manually tuned λ with respect to Nmse. The optimization problem has then been solved using a Cvx-based implementation [56, 57] provided by the authors of [118]. As the number of parameters to estimate increases

with the number of observed data points N, we observed an exponential rise in the runtime of this algorithm. For $N > 650$, the Monte Carlo simulations required to evaluate the NMSE became intractable and for $N \geq 1000$ the memory requirement became infeasible. Therefore this method was only tested on smaller data sets.

The last step in the SNR-LS approach is to divide the regressor space into regions. The authors of [118] suggest using e.g. a support vector machine (SVM), but note that such an approach is not suitable for more complicated regions. We found that the SVM approach does not always yield the desired number of regions. Therefore we opted for using the more general nearest neighbor classifier [16].

The affine ARX models have been estimated by the standard least-squares method as discussed in Section 2.5.2.

4.6.2 A Hammerstein system

Consider the system

$$y(t) = -0.5y(t-1) - 0.1y(t-2) + v(t-1) + e(t), \tag{4.16}$$

where $v(t)$ is a saturated version of $u(t)$,

$$v(t) = \begin{cases} 1 & \text{if } u(t) \geq 1 \\ u(t) & \text{if } -1 \leq u(t) \leq 1 \\ -1 & \text{if } u(t) < -1 \end{cases} \tag{4.17}$$

Here $(n_a^0, n_b^0, n_r^0) = (2, 1, 3)$. This type of system, with a static nonlinear block followed by a linear dynamic block, is commonly referred to as a Hammerstein system. The input $u(t)$ was a zero-mean white Gaussian process with variance 4, and the process noise $e(t)$ was white Gaussian with variance 0.04. This same setup was used in [118] and [115]. The system was identified using RILL, SNR-LS, and the affine ARX model. The model orders n_a, n_b where chosen equal to the true model orders for all methods.

For RILL, we used $n_r = 81$ regions for which the differences extend vertically, see Section 4.6.1.

For the SNR-LS method, we let $\ell = 8$ in (4.10), as in the corresponding example found in [118]. For the regularization weight, we chose $\lambda = 0.08$ which produced a lower NMSE than the value used in [118].

The NMSE was computed for N equal to 250, 500 and 1000. The results are shown in Table 4.1. As noted in 4.6.1, we were unable to evaluate SNR-LS for $N \geq 650$. The ARX model is, as expected, outperformed by both SNR-LS and RILL. For RILL, no particular tuning was used except for choosing the direction of the differences. Nevertheless, it performs better than SNR-LS. Moreover, if it is known that the system

Table 4.1: NMSE in Example 4.6.2.

N	250	500	1000
ARX	4.96	4.91	4.87
SNR-LS	1.76	1.57	–
RILL	1.61	1.33	1.25

has a Hammerstein structure then this prior knowledge can be exploited by RILL by only partitioning the regressor space along the u-dimension. Using $n_r = 81$ as above, the NMSE of RILL is then reduced to 1.14 already for $N = 500$.

4.6.3 A piecewise affine ARX system

For the Hammerstein system in Example 4.6.2 the poles in each region of the regressor space are the same. By contrast, consider the following PWARX system,

$$y(t+1) = \begin{cases} y(t) - 0.5y(t-1) + 0.5v(t) & \text{if } y(t) \leq 0.3 \\ 1.2y(t) - 0.35y(t-1) + 0.15v(t) & \text{if } y(t) > 0.3 \end{cases} \tag{4.18}$$

$$y(t) = \begin{cases} y(t-1) - 0.5y(t-2) & \text{if } y(t-1) \leq 0.3 \\ \quad +0.5v(t-1) + e(t), & \\ 1.2y(t-1) - 0.35y(t-2) & \text{if } y(t-1) > 0.3 \\ \quad +0.15v(t-1) + e(t) & \end{cases} \tag{4.19}$$

where $v(t)$ is again a saturated version of $u(t)$,

$$v(t) = \begin{cases} 0.8 & \text{if } u(t) \geq 0.8 \\ u(t) & \text{if } -0.8 \leq u(t) \leq 0.8 \\ -0.8 & \text{if } u(t) < -0.8 \end{cases}$$

Here $(n_a^0, n_b^0, n_r^0) = (2, 1, 6)$. Note that both linear subsystems in (4.19) have a static gain equal to one, but the poles are real for $y(t) \geq 0.3$ and complex when $y(t)$ goes below 0.3. In the simulations, $e(t)$ was chosen as white Gaussian noise with variance 0.01. The input signal $u(t)$ was chosen an RS(1.1) signal, as described in the last paragraph of Section 3.5.1.

The model orders n_a, n_b where chosen equal to the true model orders for all three identification methods. For RILL, we used $n_r = 81$ linearization points, for which the differences extend horizontally, see Section 4.6.1. For the SNR-LS method we let $\ell = 8$ in (4.10), and tuned the regularization weight to $\lambda = 0.05$.

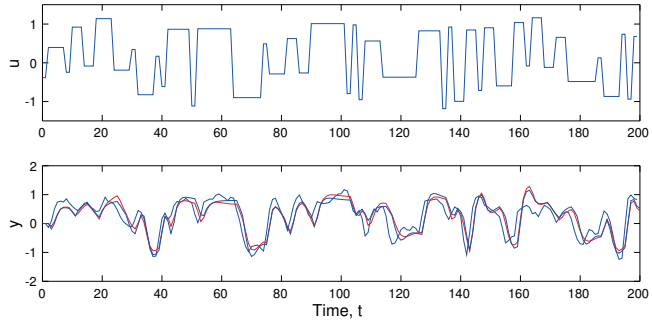


Figure 4.2: A input-output realization of (4.19) with noise (blue dashed), without noise (blue solid); also the output of the model identified by RILL using $N = 500$ samples (red).

Table 4.2: NMSE in Example 4.6.3.

N	250	500	1000
ARX	2.55	2.17	2.00
SNR-LS	1.25	1.18	–
RILL	1.18	1.10	1.05

The results for N equal to 250, 500 and 1000 are shown in Table 4.2. As noted in Section 4.6.1, we were unable to evaluate SNR-LS for $N \geq 650$. As in Example 4.6.2, the affine ARX model is outperformed by both SNR-LS and the RILL. Similarly, RILL performs better than SNR-LS. Note that in both examples we obtain similar performance as SNR-LS with $N = 500$ using only $N = 250$ data samples.

Figure 4.2 shows a realization of (4.19) together with the model output using the parameters identified by RILL from $N = 500$ samples. For the sake of clarity, we also show the same output realization when there is no process noise. It can be seen that the identified model follows the noise-free output quite well.

4.6.4 A pick-and-place machine

In this example, the data from the pick-and-place machine described in Example 3.5.4 is used.

The order of the PWARX model was set to $n_a = 2$ and $n_b = 2$ as in [118] and for RILL we used $n_r = 81$ linearization points, for which the differences extend horizontally, cf. Section 4.6.1. The input/output data are shown, together with the output of the identified model, in Figure 4.3. The fit to the validation data was 79.4% for RILL, which

is slightly better than the one of 78.6% reported in [118]. The result for an ARX-model of the same order was 73.1%. These result can also be compared to the ones in Table 3.2. In Figure 4.4, the prediction error of the model identified by RILL is compared with the identified ARX-model.

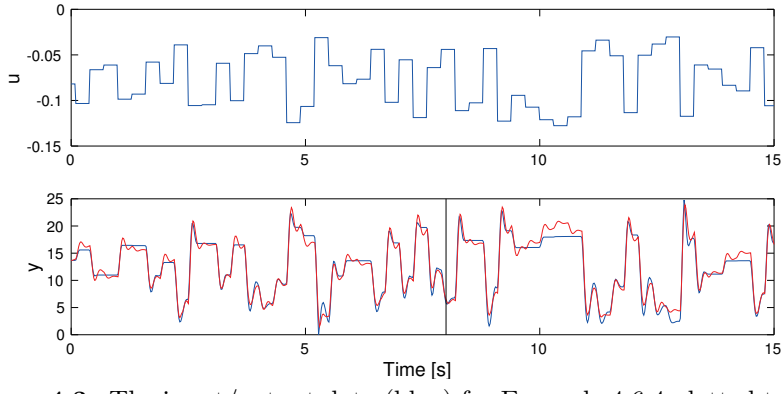


Figure 4.3: The input/output data (blue) for Example 4.6.4 plotted together with the output of the model (red) identified by RILL. The system was identified using the first 8 seconds of data.

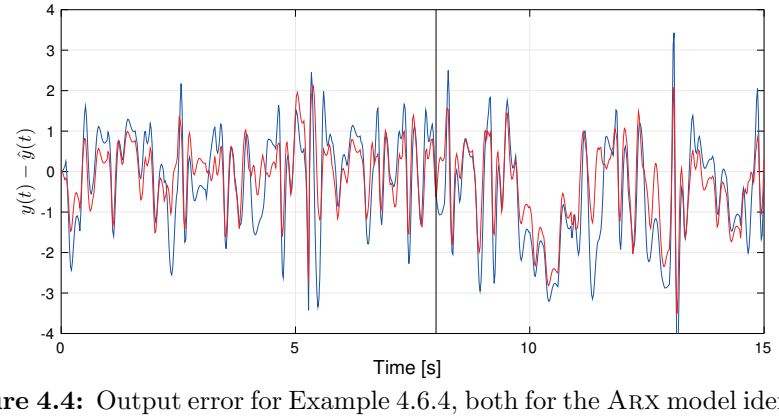


Figure 4.4: Output error for Example 4.6.4, both for the ARX model identified using LS (blue) and the PWARX model identified by RILL (red).

4.6.5 A tank process

In this example, the same tank process as in Example 3.5.3 is considered, but here we only use the measured data from the lower tank.

The identification was performed using $n_a = 4$ and $n_b = 2$ and $n_r = 81$ linearization points, for which the differences extend horizontally,

cf. Section 4.6.1. The first 1250 samples where used for identification, and the last 1250 samples for validation. The fit to the validation data was 86.9% for RILL, which can be compared to the fit of 77.4% achieved with an affine ARX model. See Figure 4.6 for a comparison of the output errors.

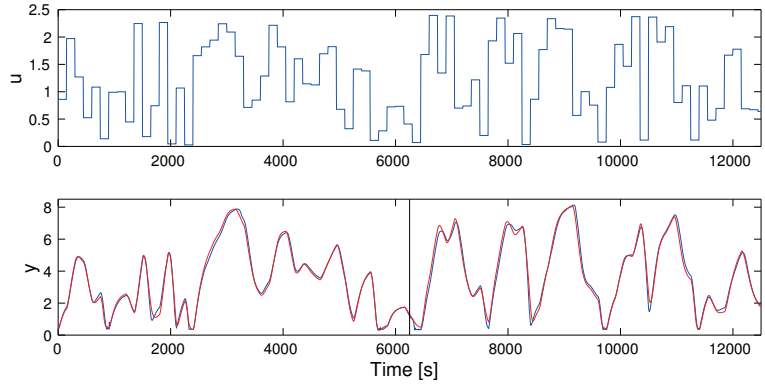


Figure 4.5: The input/output data (blue) for Example 4.6.5 plotted together with the output of the model identified by the RILL (red). The system was identified using the first 6250 seconds of data.

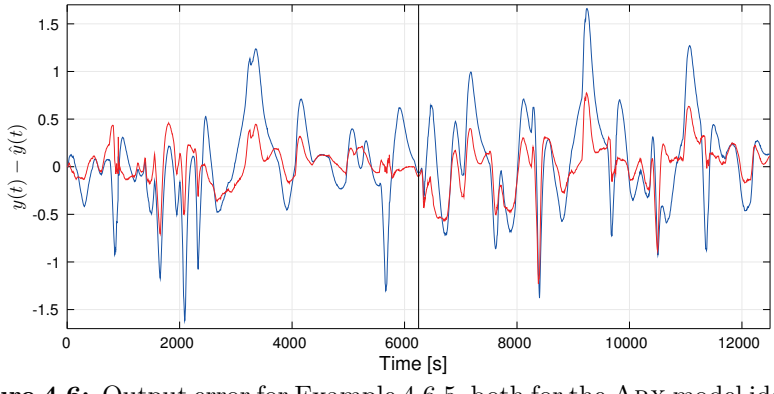


Figure 4.6: Output error for Example 4.6.5, both for the ARX model identified using LS (blue) and the PWARX model identified by RILL (red).

Chapter 5

Identification of Hammerstein models

This chapter considers recursive identification of Hammerstein models. A method based on the recursive Gauss-Newton method in Algorithm 2 is developed. The convergence properties of the algorithm are analyzed by application of Ljung's associated differential equation method. It is shown that the algorithm, under some conditions, converges to a stable stationary point of the associated differential equation. General conditions for local convergence to the true parameter vector are given, and the cases with piecewise affine and polynomial nonlinearities are treated in detail.

5.1 Introduction

In this chapter, we consider identification of the single-input/single-output (Siso) Hammerstein model. A common approach is to express the static nonlinearity as a linear combination of basis functions, as in (2.27), and then use overparameterization in order to transform the Hammerstein model into a multiplie-input/single-output linear model, see for example [23]. If the noise is correlated, instrumental-variable methods can be used to construct a consistent estimator [144]. In [17], overparameterization is combined with pseudo-inverse techniques to construct a recursive estimator for the original parameters. Another approach is the (non-recursive) iterative method [116], for which the convergence properties are studied in [6]. Common for all these methods is that the static nonlinearity is assumed to be a linear combination of known basis functions. In [25] and [167], a nonparametric model is used for the nonlinearity and a consistent estimator is constructed.

However, the method only works if the input is white, which is restrictive in practice.

The algorithm considered in this chapter also makes use of a basis expansion, but it is based on direct optimization of the SISO Hammerstein model, as in, e.g., [39] and [40]. By avoiding overparamterization we get fewer parameters to estimate, which can be very useful when the algorithm is applied to small data sets. The proposed algorithm is a recursive prediction error method (RPEM), based on the Gauss-Newton approximations, as in Algorithm 2. As such the algorithm also keeps the computational complexity low, as the running time is linear in the number of data samples.

When a new recursive identification algorithm is designed, it is also of interest to establish certain properties. Among these identifiability and convergence are of particular importance. For nonlinear models, it is in general difficult to say anything about this. However, there are tools for convergence analysis that are applicable to recursive identification algorithms, in particular Ljung's associated differential equation method [86, 91]. This method ties the convergence of the recursive identification algorithm to the stability of the associated ordinary differential equation (ODE). More specifically, local convergence of the algorithm follow if the ODE is locally stable, and global convergence is implied if the associated ODE is globally Lyapunov stable. This method has been applied to, e.g., Wiener models [162] and polynomial state-space models [147]. In this chapter such an analysis is carried out for the RPEM using the Hammerstein model, and conditions under which the algorithm is guaranteed to converge to a stable stationary point of the associated ODE are established. Furthermore, a detailed analysis of the case when the data is generated by a Hammerstein model proves local convergence to the true parameter vector under relatively mild assumptions. These assumptions include restrictions on the parameterization that are needed for identifiability, and that the input signal must be exciting in both frequency and amplitude, cf. Section 2.2. Besides giving theoretical insights into the behavior of the RPEM, this analysis provides guidelines for the user applying the method, cf. [162].

The chapter is organized with the development of the model and algorithm in Section 5.2-5.3. The convergence analysis appears in Section 5.4, and choices of the static nonlinearity are discussed in Section 5.5. In Section 5.6 the proposed method is evaluated on numerical examples. Finally, proofs and a detailed discussion about the implementation of the algorithm appear in the appendices.

5.2 The Hammerstein model

In this chapter we consider the SISO Hammerstein model, that is a model where the input signal $u(t)$ goes through a static nonlinear function $f(\cdot)$ before it enters a linear dynamical model. In this chapter we consider the linear model to be of the output error type, as in (2.17) with $d_i = 0$. The model can thus be expressed as,

$$A(q)\hat{y}(t|\theta) = B(q)f(u(t), \theta_n). \tag{5.1}$$

where $u(t)$ is the input signal, and $\hat{y}(t|\theta)$ is the predicted output signal. In Figure 5.1, the block structure of the Hammerstein model is illustrated. Note that the intermediate signal $u_n(t)$ is not available for measurement.

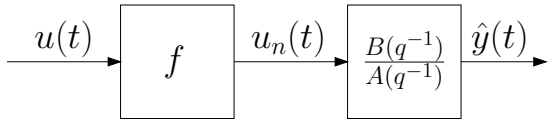


Figure 5.1: The Hammerstein model.

The linear block is modeled using the polynomials $A(q)$ and $B(q)$, which are assumed to be given as

$$B(q) = b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b}, \tag{5.2}$$

$$A(q) = 1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a}. \tag{5.3}$$

Furthermore, the parameter vector $\bar{\theta}$ for the model is partitioned as

$$\bar{\theta} = \begin{bmatrix} \bar{\theta}_n^\top & \bar{\theta}_\ell^\top \end{bmatrix}^\top, \tag{5.4}$$

where

$$\bar{\theta}_\ell = \begin{bmatrix} \bar{\theta}_a^\top & \bar{\theta}_b^\top \end{bmatrix}^\top, \tag{5.5}$$

$$\bar{\theta}_a = \begin{bmatrix} a_1 & \cdots & a_{n_a} \end{bmatrix}^\top, \tag{5.6}$$

$$\bar{\theta}_b = \begin{bmatrix} b_1 & \cdots & b_{n_b} \end{bmatrix}^\top, \tag{5.7}$$

and $\bar{\theta}_n$ are the parameters of the static nonlinearity. In order to handle the static nonlinearity, we consider it to be a linear combination of some basis functions as in (2.27), that is

$$f(u, \theta_n) = \sum_{i=1}^{n_k} k_i f_i(u) = \bar{\theta}_n^\top F(u), \tag{5.8}$$

where

$$\bar{\theta}_n = \begin{bmatrix} k_1, \ldots, k_{n_k} \end{bmatrix}^\top, \tag{5.9}$$

$$F(u) = \begin{bmatrix} f_1(u) & \cdots & f_{n_k}(u) \end{bmatrix}^\top. \tag{5.10}$$

Using these notations, (5.1) can be written as pseudolinear regression

$$\hat{y}(t|\theta) = \bar{\theta}_\ell^\top \varphi(t,\theta), \tag{5.11}$$

where

$$\varphi(t,\theta) = [-\hat{y}(t-1|\theta) \quad \cdots \quad -\hat{y}(t-n_a|\theta) \\ f(u(t-1),\theta_n) \quad \cdots \quad f(u(t-n_b),\theta_n)]^\top. \tag{5.12}$$

Because of the cascade structure in (5.1), it is not possible to separate the gain in the linear block from the gain in the static nonlinearity. That is, the transfer function $G(q) = B(q)/A(q)$ together with the static nonlinearity $f(\cdot)$, gives the same input-output behavior as the transfer function $G(q)/\alpha$ with the static nonlinearity $\alpha f(\cdot)$ for any constant $\alpha \neq 0$. Hence, for the purpose of identification, it is reasonable to fixate the gain in one of the blocks in some way. There are several ways to accomplish this. For example, b_1 could be fixed equal to 1. In this case, the linear block is determined uniquely by the zeros and poles of the transfer function, and the gain of the model can be adjusted by the static nonlinearity. The drawback of this approach is that it is not useful when there is an unknown time-delay in the system.

Another approach is to fixate the gain in the static nonlinearity. For example, it could be assumed that $f(u,\theta_n)$ has a given constant slope in some interval I_r, i.e.,

$$\frac{\partial}{\partial u} f(u,\theta_n) = k_r, \quad \text{if } u \in I_r, \tag{5.13}$$

where k_r is a known constant. This approach was also used in [160] and [161].

In this chapter it is assumed that one element in the vector $\bar{\theta}$ is fixed, and known beforehand. Hence, it follows that the vector θ of unknown parameters is

$$\theta = I_o \bar{\theta}, \tag{5.14}$$

where I_o is a matrix that removes one row, i.e., the identity matrix with one row removed. In Section 5.4.5 it will be seen that in order to guarantee local convergence, the fixated parameter should be either b_i or k_i for some i.

5.3 The recursive identification algorithm

In this section a recursive prediction error algorithm (RPEM) for the model structure in (5.1)-(5.8) is derived by minimization of the criterion

$$V(\theta) = \frac{1}{2} \mathrm{E}\, \varepsilon^2(t,\theta), \tag{5.15}$$

where

$$\varepsilon(t,\theta) = y(t) - \hat{y}(t|\theta), \tag{5.16}$$

is the prediction error and $y(t)$ is the measured output of the system. Also define the set of all parameter vectors θ such that the linear part is stable as

$$\mathcal{D}_s = \left\{ \theta \,\middle|\, \frac{1}{A(q)} \text{ is asymptotically stable} \right\}.$$

For the convergence analysis, it must also be ensured that the estimates stay in a compact set. Therefore, let $\mathcal{D}_{\mathcal{M}}$ be a compact subset of $\mathcal{D}_s$.

We now apply the recursive Gauss-Newton algorithm described in Section 2.5.3. From (2.50), it can be seen that the negative gradient of the prediction error is given by

$$\psi(t,\theta) = \left[\frac{d}{d\theta}\hat{y}(t|\theta)\right]^\top = I_o \left[\frac{d}{d\bar{\theta}}\hat{y}(t|\theta)\right]^\top = I_o \begin{bmatrix} \overline{\psi}_n(t,\theta) \\ \overline{\psi}_l(t,\theta) \end{bmatrix}, \tag{5.17}$$

where

$$\overline{\psi}_n(t,\theta) = \left[\frac{d}{d\bar{\theta}_n}\hat{y}(t|\theta)\right]^\top, \quad \overline{\psi}_\ell(t,\theta) = \left[\frac{d}{d\bar{\theta}_\ell}\hat{y}(t|\theta)\right]^\top.$$

As in, e.g., [91], it can be shown that

$$\overline{\psi}_\ell(t,\theta) = \frac{1}{A(q)}\varphi(t,\theta), \tag{5.18}$$

and for the static nonlinearity we have

$$\overline{\psi}_n(t,\theta) = \frac{B(q)}{A(q)}\left[\frac{d}{d\bar{\theta}_n} f(u(t),\theta_n)\right]^\top = \frac{B(q)}{A(q)}F(u(t)), \tag{5.19}$$

where the last equality follows from (5.8).

Lemma 5.1. *For a given $\theta \in \mathcal{D}_s$, the negative gradient and the model output can be generated by a linear state-space model on the form*

$$\xi(t+1,\theta) = A(\theta)\xi(t,\theta) + B(\theta)F(u(t)), \tag{5.20}$$

$$\begin{bmatrix} \hat{y}(t|\theta) \\ \psi(t,\theta) \end{bmatrix} = C(\theta)\xi(t,\theta), \tag{5.21}$$

where $A(\theta)$ has all eigenvalues strictly inside the unit circle.

Proof. See Appendix 5.B.1. □

The recursive Gauss-Newton type algorithm for the Hammerstein model is thus given by Algorithm 2, with the matrices $A(\theta)$, $B(\theta)$ and $C(\theta)$ given by Lemma 5.1.

5.3.1 Implementation

The formulation of Algorithm 2 is suitable for the convergence analysis that will be carried out in Section 5.4. However, there are some tricks that can be used for implementing it on a computer. First, in Algorithm 2, the inverse of $R(t)$ is needed in each time step. However, as in recursive least-squares, we can avoid computing this inverse explicitly. Let

$$P(t) \triangleq \gamma(t) R^{-1}(t).$$

By applying the matrix inversion lemma to (2.52), it can then be seen that

$$P(t) = \frac{1}{\lambda(t)} \left(P(t-1) - P(t-1)\psi(t)\psi^{\top}(t)P(t-1)/S(t) \right), \tag{5.22}$$

$$S(t) = \psi^{\top}(t)P(t-1)\psi(t) + \lambda(t), \tag{5.23}$$

$$\lambda(t) = \frac{\gamma(t-1)}{\gamma(t)} \left(1 - \gamma(t)\right). \tag{5.24}$$

Note that, in the recursion for $P(t)$, we only need to compute the inverse for the scalar variable $S(t)$. While this recursion usually works well for low dimensions of the parameter vector, it can be sensitive to round-off errors especially for large dimensions of the parameter vector. In these cases alternatives, such as the U-D factorization or square-roots algorithms, can be used – see [91] for more details.

The sequence $\lambda(t)$ is typically called the forgetting factor. Note that, for $\gamma = 1/t$, we get $\lambda(t) = 1$. As discussed in Section 2.5.3, it might be useful to choose $\gamma(t) > 1/t$ in order to put more weight on recent estimate, and this corresponds to letting $\lambda(t) < 1$. On the other hand it is desirable that $t\gamma(t) \to 1$ as $t \to \infty$, which corresponds to $\lambda(t) \to 1$. A common choice for the forgetting factor is to let [91],

$$\lambda(t) = \lambda_o \lambda(t-1) + (1 - \lambda_o),$$

so that $\lambda(t)$ grows exponentially from $\lambda(0)$ to 1 with the rate $\lambda_o < 1$. Here λ_o and $\lambda(0)$ are design variables. In many practical applications, the numerical values

$$\lambda_o = 0.99, \quad \lambda(0) = 0.95,$$

have proven useful [91]. A detailed discussion on the implementation of the RPEM is provided in Appendix 5.A, where a version of the method that is straightforward to implement in, e.g., MATLAB is presented.

5.4 Convergence analysis

The convergences properties of the RPEM are analysed using the associated differential equation approach described in [86] and [91].

In [86], it was shown that Algorithm 2 is related to the following ODE:

$$\frac{d}{d\tau}\theta_D(\tau) = R_D^{-1}(\tau) f_A(\theta_D(\tau)), \tag{5.25}$$

$$\frac{d}{d\tau}R_D(\tau) = G_A(\theta_D(\tau)) - R_D(\tau), \tag{5.26}$$

where

$$f_A(\theta) \triangleq \lim_{t\to\infty} \mathrm{E}[\psi(t,\theta)\varepsilon(t,\theta)], \tag{5.27}$$

$$G_A(\theta) \triangleq \lim_{t\to\infty} \mathrm{E}[\psi(t,\theta)\psi^\top(t,\theta)]. \tag{5.28}$$

Note that both $f_A(\theta)$ and $G_A(\theta)$ are computed using a fixed θ.

Intuitively, the relation between the algorithm and the ODE is that, if some conditions on the data and algorithm hold, then only stable stationary points of (5.27)-(5.28) are possible convergence points of the RPEM. Also, the trajectories of $\theta_D(\tau)$ are the asymptotic paths of the estimates $\hat{\theta}(t)$.

In order to use the associated ODE approach, it is required to represent the RPEM on state-space form, this is performed in Section 5.4.1. In Section 5.4.2, conditions on data generation that ensure the applicability of the results in [86] are presented, and in Section 5.4.3 it is shown that these conditions ensure that the estimates converge to a stable stationary point of (5.25)-(5.26) or to the boundary of $\mathcal{D}_\mathcal{M}$. This provides the foundation for the detailed analysis of the Hammerstein identification algorithm that follows in Section 5.4.4-5.4.5. In these sections, conditions that ensure that the true parameter vector is a possible convergence point are given. This requires that detailed conditions on the parametrization and signals are introduced, as in, e.g. [162]. In particular, this provides new insight on why it is important to excite the system well in both amplitude and frequency, cf. [163]. Finally, the convergence analysis is summarized in Section 5.4.6.

5.4.1 State-space representation of the RPEM

In this section, the algorithm in Section 5.3 is reformulated in a way that corresponds to the general structure in [86]. Define

$$\hat{x}(t) \triangleq \begin{bmatrix} \hat{\theta}^\top(t) & (\operatorname{vec} R(t))^\top \end{bmatrix}^\top. \tag{5.29}$$

Note that, given $\hat{x}(t-1)$ and $\xi(t)$, we can compute $\varepsilon(t)$, $\psi(t)$ and $R(t)$. Define

$$Q(t, \hat{x}(t-1), \xi(t)) \triangleq \begin{bmatrix} \mu(t) R^{-1}(t)\psi(t)\varepsilon(t) \\ \mu(t) \operatorname{vec}(\psi(t)\psi^\top(t) - R(t))) \end{bmatrix},$$

where $\mu(t) = t\gamma(t)$. Then it follows immediately that Algorithm 2 can be formulated as in [86], that is

$$\hat{x}(t) = \hat{x}(t-1) + \frac{1}{t}Q(t, \hat{x}(t-1), \xi(t)), \tag{5.30}$$

$$\xi(t+1) = A(\hat{\theta}(t))\xi(t) + B(\hat{\theta}(t))F(u(t)). \tag{5.31}$$

5.4.2 Conditions on the algorithm and the data

For the purpose of convergence analysis, it is assumed that $y(t)$ is generated by

$$\bar{y}(t) = \frac{B_o(q)}{A_o(q)} f(u(t), \theta_n^o), \tag{5.32}$$

$$y(t) = \bar{y}(t) + w(t), \tag{5.33}$$

where $w(t)$ is a measurement disturbance and

$$B_o(q) = b_1^o q^{-1} + \cdots + b_{n_b^o}^o q^{-n_b^o}, \tag{5.34}$$

$$A_o(q) = 1 + a_1^o q^{-1} + \cdots + a_{n_a^o}^o q^{-n_a^o}. \tag{5.35}$$

It is further assumed that $n_a^o \leq n_a$, $n_b^o \leq n_b$ and that the true parameter vector θ^o is padded with zeros in order to get the same dimension as θ. Finally, it is assumed that $\theta^o \in \mathcal{D}_{\mathcal{M}}$.

In order to analyse Algorithm 2, a number of regularity conditions are needed. The conditions given for the RPEM algorithm in [91] are

- M1 $\mathcal{D}_{\mathcal{M}}$ is a compact subset of $\mathbb{R}^d$, such that $\theta \in \mathcal{D}_{\mathcal{M}} \Rightarrow A(\theta)$ has all eigenvalues strictly inside the unit circle.
- M2 The matrices $A(\theta)$, $B(\theta)$ and $C(\theta)$ in (5.20)-(5.21) are continuously differentiable w.r.t θ for $\theta \in \mathcal{D}_{\mathcal{M}}$.
- R1 The generation of the matrix $R(t)$ is such that for all t, $R(t)$ is symmetric and $R(t) \geq \delta I$, for some $\delta > 0$.
- G1 $\lim_{t\to\infty} t\gamma(t) = \mu > 0$.
- A3 For fixed $\theta \in \mathcal{D}_{\mathcal{M}}$ the limits in (5.27)-(5.28), as well as

$$\overline{V}(\theta) \triangleq \lim_{t\to\infty} \mathrm{E}[\varepsilon^2(t, \theta)] \tag{5.36}$$

  exist.
- S1 For each t, s with $t \geq s$, there exists a random vector $z_s^o(t)$ (with $z_s^o(s) = 0$) that belongs to the σ-algebra generated by z^t but is independent of z^s, such that

$$\mathrm{E}\,\|z(t) - z_s^o(t)\|^4 < C\lambda^{t-s},\ C < \infty,\ |\lambda| < 1. \tag{5.37}$$

  Here z^t is the data set made up of $z(0), \ldots, z(t)$, where

$$z(t) = \begin{bmatrix} y(t) \\ F(u(t)) \end{bmatrix}.$$

Remark 5.1. *Condition A3 differs slightly from A3 given in [91], since the operation*

$$\overline{\mathrm{E}}[f(t)] = \lim_{t\to\infty} \frac{1}{t}\sum_{k=1}^{t} \mathrm{E}[f(k)]$$

used in [91], is changed to

$$\lim_{t\to\infty} \mathrm{E}[f(t)].$$

However, it is easy to see that the condition used here is stronger than the condition used in [91], cf. [162].

The conditions above might be a bit hard to verify in practice, but when $y(t)$ is generated by a Hammerstein system, as in (5.32)-(5.33), these conditions can be replaced by a set of more straightforward conditions, as shown by the following lemma.

Lemma 5.2. *Assume that $y(t)$ is generated by* (5.32)-(5.33) *with $\theta^o \in D_{\mathcal{M}}$, and that the following conditions are satisfied.*

- **C1** $u(t)$ is a strictly stationary process.
- **C2** $w(t)$ is a strictly stationary process that is uncorrelated with $u(t)$, and $\mathrm{E}\, w(t) = 0$.
- **C3** $f(u(t), \theta_n)$ has bounded fourth order moments.
- **C4** The disturbance $w(t)$ satisfies a condition like S1, i.e.,

$$\mathrm{E}\,\|w(t) - w_s^o(t)\|^4 \leq C\lambda^{t-s},\ C < \infty,\ |\lambda| < 1.$$

- **C5** The gain sequence $\gamma(t) = \mu(t)/t$ and $\lim_{t\to\infty} \mu(t) = \mu$.
- **C6** The generation of the matrix $R(t)$ is such that for all t, $R(t)$ is symmetric and $R(t) \succeq \delta I$, for some $\delta > 0$.

Then M1, M2, R1, G1, A3 and S1 hold.

Proof. See Appendix 5.B.2. □

Remark 5.2. *Condition C4 is satisfied for example when $w(t)$ is an ARMA process.*

Remark 5.3. *In practice, it is possible to modify the updates of $R(t)$, so that a small positive definite matrix is added to $R(t)$, when needed. In this way C6 will be satisfied, cf. [160].*

5.4.3 Global convergence

Given the conditions in Section 5.4.2, the tools of analysis in [86] and [91] can now be applied to prove the following convergence result for the RPEM in Section 5.3.

Theorem 5.1. *Assume that $y(t)$ is generated by (5.32)-(5.33) with $\theta^o \in \mathcal{D}_\mathcal{M}$ and that C1-C6 are satisfied. Also assume that there exists a bounded subsequence of $\{\xi(t)\}$ generated by (5.31). Then $\{\hat{\theta}(t)\}$ converges with probability one either to the set*

$$\mathcal{D}_c = \{\theta \,|\, f_A(\theta) = 0\},$$

or to the boundary of $\mathcal{D}_\mathcal{M}$ as $t \to \infty$.

Proof. This is Theorem 4.3 in [91]. Lemma 5.2 shows that all conditions in [91], except Cr2 and Cr3, follow from C1-C6. That Cr2 and Cr3 holds for an RPEM with criterion function (5.15) is shown on page 164 in [91]. □

Since

$$\frac{d}{d\theta}\overline{V}(\theta) = -f_A^\top(\theta),$$

it follows from Theorem 5.1 that $\hat{\theta}(t)$ converges to a critical point of the criterion function (5.36), or to the boundary of $\mathcal{D}_\mathcal{M}$.

Next we show that the algorithm only can converge to stable stationary points of (5.25)-(5.26). This requires that the following condition holds.

C7 $\|z(t)\| \leq C < \infty$ w.p.1, where $z(t) = \begin{bmatrix} y(t) \\ F(u(t)) \end{bmatrix}$.

This condition states that the data vector is bounded, which is satisfied for example when the data is generated by an asymptotically stable Hammerstein process, provided that $F(u(t))$ and the noise is bounded.

Theorem 5.2. *Consider the* RPEM *in Section 5.3 subject to conditions C1-C7. Let $\mathcal{B}(x^*, \rho)$ denote a ρ-neighbourhood of*

$$x^* = \begin{bmatrix} \theta^* \\ \operatorname{vec} R^* \end{bmatrix}.$$

Assume that $\theta^* \in \mathcal{D}_\mathcal{M}$, and that

$$Prob(\hat{x}(t) \to \mathcal{B}(x^*, \rho)) > 0$$

for all $\rho > 0$. Furthermore assume that

- *$Q(t, x^*, \xi^*)$ has a covariance matrix bounded from below by a strictly positive definite matrix, and that*
- *$\mathrm{E}\, Q(t, x, \xi(t))$ is continuously differentiable with respect to x in a neighbourhood of x^* and that the derivatives converge uniformly in this neighborhood as t tends to infinity.*

Then (θ^*, R^*) *is a stable stationary point of* (5.25)-(5.26).

Proof. In [86], a set of conditions that ensure the result of this theorem are given. These conditions can be verified using the same techniques as in [162]. □

The result in Theorem 5.2 may seem a bit technical, but a simple interpretation is that it corresponds to Result 4.3 in [91], i.e., the algorithm in Section 5.3 can only converge to values (θ^*, R^*) that are stable stationary points of (5.25)-(5.26). The results in this section can be summarized as in the following theorem.

Theorem 5.3. *Consider the algorithm in Section 5.3 subject to conditions C1-C7, as well as the assumptions in Theorem 5.2. Then* $\hat{\theta}(t)$ *converges to either a local minimum of the criterion function* (5.36)*, or to the boundary of* $\mathcal{D}_{\mathcal{M}}$.

Proof. Since $\frac{d}{d\theta}\overline{V}(\theta) = -f_A^\top(\theta)$ it follows from Theorem 5.1 that $\hat{\theta}(t)$ converges to a critical point of (5.36), or to the boundary of $D_{\mathcal{M}}$. That the critical point must be a local minimum follows from Theorem 5.2. □

5.4.4 Local convergence

In Section 5.4.3, we saw that, given some conditions, the RPEM can only converge to stable stationary points of (5.25)-(5.26). In this section the goal is to give some conditions that ensure that the true parameter vector θ^o is one of these stable stationary points. As in the proof of Theorem 5.3, it can be shown that this is equivalent to θ^o being a local minimum of the criterion (5.36).

Note that (5.25)-(5.26) depends on the asymptotic behavior of the negative gradient $\psi(t, \theta)$. In order to find a convenient expression for $\psi(t, \theta)$, let

$$M(\theta) \triangleq I_o \begin{bmatrix} h^\top \otimes I_{n_k} \\ S(-B, A) \otimes \overline{\theta}_n^\top \end{bmatrix}, \tag{5.38}$$

where h contains the coefficients of the polynomial

$$H(q^{-1}) \triangleq A(q^{-1})B(q^{-1}) = h_1 q^{-1} + \cdots + h_{n_a+n_b} q^{-n_a-n_b},$$

and $S(-B, A)$ is the Sylvester matrix,

$$S(-B,A) = \begin{bmatrix} 0 & -b_1 & \cdots & -b_{n_b} & & 0 \\ \vdots & \ddots & \ddots & & \ddots & \\ 0 & \cdots & 0 & -b_1 & \cdots & -b_{n_b} \\ 1 & a_1 & \cdots & a_{n_a} & & 0 \\ & \ddots & \ddots & & \ddots & \\ 0 & & 1 & a_1 & \cdots & a_{n_a} \end{bmatrix}. \tag{5.39}$$

Also introduce

$$\mathbf{F}_n(t) \triangleq \begin{bmatrix} F(u(t-1)) \\ \vdots \\ F(u(t-n)) \end{bmatrix}, \quad \text{and} \quad v_n(t,\theta) = \frac{1}{A^2(q)}\mathbf{F}_n(t). \tag{5.40}$$

Lemma 5.3. *The negative gradient is given by*

$$\psi(t,\theta) = M(\theta)v_{n_a+n_b}(t,\theta).$$

Proof. See Appendix 5.B.3. □

Now an expression for $f_A(\theta)$ in (5.27) can be found.

Lemma 5.4. *Assume that C1-C2 hold, and that $y(t)$ is generated by* (5.32)-(5.33) *with $\theta^o \in \mathcal{D}_{\mathcal{M}}$. Then,*

$$f_A(\theta) = \lim_{t\to\infty} \mathrm{E}\left[\psi(t,\theta)\tilde{\psi}^\top(t,\theta)\right](\theta^o - \theta), \tag{5.41}$$

where

$$\tilde{\psi}(t,\theta) \triangleq I_o \frac{1}{A_o(q)}\begin{bmatrix} B_o(q)F(u(t)) \\ \varphi(t,\theta) \end{bmatrix}. \tag{5.42}$$

Proof. See Appendix 5.B.4. □

Note that setting $\theta = \theta^o$ in (5.41) and (5.42) give $\tilde{\psi}(t,\theta^o) = \psi(t,\theta^o)$ and $f_A(\theta^o) = 0$.

Lemma 5.5. *If C1-C2 hold and $G_A(\theta^o)$ is positive definite, then θ^o is a locally stable stationary point of* (5.25)-(5.26).

Proof. The fact that the point is stationary follows from Lemma 5.4. On page 178 in [91], it is shown that the point is locally stable if $G_A(\theta^o)$ is invertible and

$$L = G_A^{-1}(\theta^o)\left[\frac{d}{d\theta}f_A(\theta)\right]_{\theta=\theta^o}$$

has all eigenvalues in the left half-plane. From Lemma 5.4, it follows that $L = -I$ when $G_A(\theta^o)$ is positive definite, so the lemma follows. □

5.4.5 Conditions on the model

In Section 5.4.4, it was shown that the RPEM is locally convergent to θ^o if $G_A(\theta^o)$ is positive definite. In this section, conditions on the model that ensure positive definiteness of $G_A(\theta^o)$ are given.

Lemma 5.6. *Assume that condition C1 holds, and let*

$$P \triangleq \mathrm{E}\left[\mathbf{F}_{n_a+n_b}(t)\mathbf{F}_{n_a+n_b}^{\top}(t)\right]. \tag{5.43}$$

Then $G_A(\theta^o) \succ 0$ *if and only if*

$$M(\theta^o)PM^{\top}(\theta^o) \succ 0.$$

Proof. See Appendix 5.B.5. □

From Lemma 5.6 it can be seen that a necessary condition for $G_A(\theta) \succ 0$ is that $M(\theta)$ has full row rank.

Lemma 5.7. *Assume that the following conditions are satisfied.*
C8 *$A_o(q)$ and $B_o(q)$ are coprime and* $\min(n_a - n_a^o, n_b - n_b^o) = 0$.
C9 *The fixed parameter in $\bar{\theta}$ is either $b_i \neq 0$, or $k_i \neq 0$ for some i.*
Then $M(\theta^o)$ has full row rank.

Proof. See Appendix 5.B.6. □

Remark 5.4. *Condition C8 and C9 deal with the parameterization of the model. These conditions are needed to secure that θ^o is a local minimum to the criterion* (5.36)*, which is a prerequisite for local convergence according to Theorem 5.2. Hence, these conditions are also related to identifiability of the model.*

Remark 5.5. *As discussed in Section 5.2, one parameter in $\bar{\theta}$ has to be fixed in order not to overparameterize the model. From the proof of Lemma 5.7, it can be seen that if we fixate one of the parameters in $A(q)$, then $M(\theta)$ will be rank deficient, and hence $G_A(\theta^o)$ is not positive definite in this case. However, if we instead decides to fixate either one of the parameters in $B(q)$ or one in $\bar{\theta}_n$, then $M(\theta^o)$ still has full row rank.*

For local convergence it is also required that the input signal is sufficiently exciting, as discussed in the next lemma.

Lemma 5.8. *Assume that conditions C1 and C8-C9 as well as one of the following two conditions hold.*

C10 The signal $F(u(t))$ is persistently exciting (pe) of order $n_a + n_b$, i.e., $\mathrm{E}\left[\mathbf{F}_{n_a+n_b}(t)\mathbf{F}_{n_a+n_b}^{\top}(t)\right]$ is positive definite.

C11 The first basis function is $f_1(u(t)) = 1$, $B_o(1) \neq 0$, and the fixed parameter in $\bar{\theta}$ is not k_1. Also, the signal $F_o(u(t))$ is pe of order $n_a + n_b$ where

$$F_o(u(t)) = \begin{bmatrix} f_2(u(t)) & \cdots & f_{n_k}(u(t)) \end{bmatrix}^{\top}. \tag{5.44}$$

Then $G_A(\theta^o)$ is positive definite.

Proof. See Appendix 5.B.7. □

The reason for dividing Lemma 5.8 into two different conditions is that C10 cannot hold when one of the basis functions in $F(u)$ is constant. In this case condition C11 can be used instead. Note that, if the static gain of the linear block is zero, i.e., if $B_o(1) = 0$, then the constant term in $f(u, \theta)$ will not affect the output asymptotically. Hence the assumption $B_o(1) \neq 0$ is needed when a constant term in the nonlinearity has to be identified.

When the input is a stationary white noise process, the conditions in Lemma 5.8 can be simplified, as shown in the following corollary.

Corollary 5.1. *Assume that $u(t)$ is a white noise process. Let $m_F = \mathrm{E}[F(u(t))]$ and $m_{F_o} = \mathrm{E}[F_o(u(t))]$.*

- *If $\mathrm{E}(F(u(t)) - m_F)(F(u(t)) - m_F)^{\top}$ is positive definite, then C10 holds.*
- *If $f_1(u(t)) = 1$, $B_o(1) \neq 0$, the fixed parameter in $\bar{\theta}$ is not k_1, and $\mathrm{E}(F_o(u(t)) - m_{F_o})(F_o(u(t)) - m_{F_o})^{\top}$ is positive definite, then C11 holds.*

Proof. See Appendix 5.B.8. □

Since the statement in Corollary 5.1 does not involve the full vector $\mathbf{F}(t)$, it can be very useful in establishing C10 or C11, as shown in the proof of Lemma 5.10.

5.4.6 Summary of the convergence analysis

In this section, several conditions on the algorithm and data have been introduced. Note that conditions C1-C4 deal with properties of the input and output signals. Condition C4 is a bit technical, but it is satisfied for example when $w(t)$ is an ARMA process. Conditions C5-C6 deal with the algorithm. In order to satisfy condition C6, it is good practice to add a small positive definite matrix to $R(t)$ to secure that it is positive definite. If the algorithm is implemented using the P-recursion

in (5.22) instead, this addition of a small positive definite matrix is done implicitly, cf. recursive least squares in Section 2.5.2.

Condition C7 states that the data vector $z(t)$ should be bounded, something that is rarely a problem in practice. As stated above, conditions C8-C9 ensure a unique parameterization of the model, while conditions C10-C11 are used to guarantee excitation.

The convergence analysis carried out in this section can be summarized by the following theorem.

Theorem 5.4. *Assume that the measured output signal $y(t)$ is generated by* (5.32)-(5.33) *with $\theta^o \in \mathcal{D}_\mathcal{M}$. If conditions C1-C7, and the assumptions of Theorem 5.2, are satisfied, then*

- *the estimate $\hat{\theta}(t)$ converges to a local minimum of the criterion* (5.36)*, or to the boundary of $\mathcal{D}_\mathcal{M}$.*

If also condition C8-C9 as well as either C10 or C11 are satisfied, then

- *the true parameter vector θ^o is a local minimum of* (5.36)*.*

5.5 The static nonlinearity

In this section, the choice of the static nonlinearity $f(u, \theta)$ in (5.1) is discussed. In order to ensure local convergence to the true parameter vector, the nonlinearity must satisfy C10 or C11 for the input signal used.

5.5.1 Polynomial nonlinearity

A simple, and commonly used, basis expansion is that of polynomials. That is, in (5.8), use

$$f_i(u) = u^{i-1}, \quad i = 1, \ldots, n_k. \tag{5.45}$$

This expansion has $f_1(u) = 1$, so condition C11 must be verified.

Lemma 5.9. *If the static nonlinearity is given by* (5.8) *with* (5.45)*, and the input signal is given by an* ARMA *process, then condition C11 holds.*

Proof. See [144]. □

5.5.2 Piecewise affine nonlinearity

As mentioned in Chapter 4, it is popular to consider the static nonlinear function to be piecewise affine. In order to define the piecewise affine

function, first choose a set of grid points

$$\text{grid} = \begin{bmatrix} u_1 & \cdots & u_{n_k} \end{bmatrix}.$$

The static nonlinearity is then assumed to have a constant slope in-between two consecutive grid points. There are several ways to parameterize this kind of function. In [160] the parameters where chosen to correspond to the function value at each grid point. Another option is to let each parameter describe the slope between two grid points. The latter approach is used here.

We also need one parameter to describe the bias of the nonlinearity. Hence, let $k_1 = f(u_1, \theta_n)$, $f_1(u) = 1$, and let $k_i, i > 1$, be the slope in the interval (u_{i-1}, u_i). Then, for $u \in [u_{i-1}, u_i]$, we get

$$f(u, \theta_n) = k_1 + k_2(u_2 - u_1) + \cdots + k_{r-1}(u_{r-1} - u_{r-2}) + k_r(u - u_{r-1}).$$

The basis functions $f_i(u)$, $i = 1, \ldots, n_k$ in (5.8) can thus be defined as

$$f_i(u) = \begin{cases} 1 & \text{if } i = 1 \\ u - u_{i-1} & \text{if } i \neq 1, \text{ and } u \in (u_{i-1}, u_i) \\ u_i - u_{i-1} & \text{if } i \neq 1, \text{ and } u \geq u_i \\ 0 & \text{otherwise} \end{cases} \tag{5.46}$$

Given this parameterization, the following result holds.

Lemma 5.10. *Consider the static nonlinearity given by* (5.8) *with* (5.46)*, and let* $I_i = [u_{i-1}, u_i]$*. Furthermore, assume that* $u(t)$ *is a white noise process, with a probability density function* $p_u(\cdot)$ *that satisfies*

$$p_u(u) \geq \delta > 0 \tag{5.47}$$

in at least one nonzero interval $[c_i, d_i] \subset I_i$*, for all* $i = 2, \ldots, n_k$*. Then condition C11 is satisfied.*

Proof. See Appendix 5.B.9. □

The condition on the probability density function in Lemma 5.10 ensures that $u(t)$ has amplitudes in all intervals of the static nonlinearity. That such a condition is needed is reasonable, since if there is no signal energy in an interval, it will be impossible to find out the slope in this interval. In fact, to estimate the slope, at least two points on the line should be considered. The condition on $p_u(u)$ makes sure that $u(t)$ can take on infinitely many different values in each interval. However, it can be seen from the proof that it would be enough to assume that $p_u(u)$ has a point mass in at least two points in each interval, cf. the analysis of [160], [161].

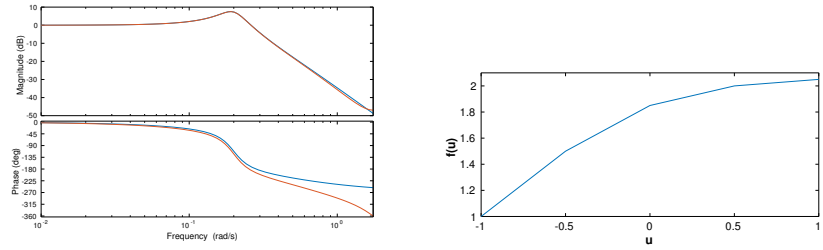


Figure 5.2: The bode plot (left) and static nonlinear block (right) of the generating system in Section 5.6.1.

5.6 Numerical examples

In this section, the RPEM for Hammerstein models is validated on several numerical examples. In Example 5.6.1-5.6.2, the case when the system generating the output is within the model structure are considered. It can be seen that the RPEM indeed converges to the true parameter vector in these examples. In order to test the algorithm in non-ideal situations, Example 5.6.3-5.6.4 consider the case when the generating system is not in the model structure considered. In these examples, the algorithm cannot converge to a "true" parameter vector, but it is shown that the identified Hammerstein model approximate both the linear dynamics and the static nonlinearity well.

5.6.1 A saturating nonlinearity

For validation of the algorithm, time data was generated by a piecewise linear continuous system, with linear block

$$G(s) = \frac{0.02}{s^3 + 0.58s^2 + 0.08s + 0.02}. \tag{5.48}$$

The continuous time system was then sampled with a sampling interval of 2 s. The Bode plot and the piecewise linear nonlinearity are shown in Figure 5.2. White Gaussian noise with standard deviation 0.01 was then added to the output.

The parameterization in Section 5.5.2 was used for the static nonlinearity, with a grid coinciding with that of the true nonlinearity and the fixed parameter was chosen as the slope in the first interval. The input was chosen as uniform white noise, thus satisfying the conditions of Lemma 5.10.

The prediction error and parameter estimates generated by the algorithm are shown in Figure 5.3. After 5000 samples, the following

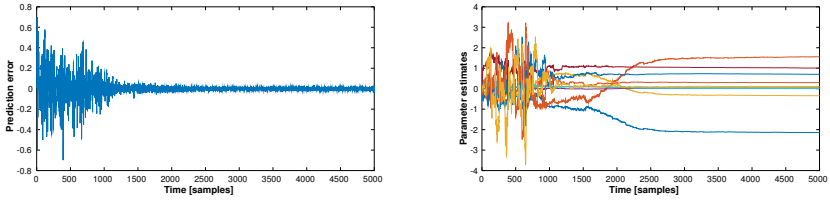


Figure 5.3: The prediction error $\varepsilon(t)$ (left) and parameter estimates $\hat{\theta}(t)$ (right) in Section 5.6.1.

parameter estimates where obtained

$$\hat{\theta}_l = \begin{bmatrix} -2.145 & 1.561 & -0.345 & 0.015 & 0.047 & 0.010 \end{bmatrix}^\top,$$
$$\hat{\theta}_n = \begin{bmatrix} 1.01 & 0.71 & 0.30 & 0.11 \end{bmatrix}^\top,$$

which can be compared to the true sampled parameters,

$$\theta_l = \begin{bmatrix} -2.153 & 1.576 & -0.352 & 0.015 & 0.047 & 0.009 \end{bmatrix}^\top,$$
$$\theta_n = \begin{bmatrix} 1.0 & 0.7 & 0.3 & 0.1 \end{bmatrix}^\top.$$

It can be concluded that the algorithm indeed converges towards the true parameter vector.

5.6.2 A nonmonotonic nonlinearity

For validation of the algorithm, data were generated by a piecewise linear continuous time system, with linear block equal to the one in [160], that is

$$G(s) = \frac{37}{s^2 + 7s + 37}. \tag{5.49}$$

The nonlinearity was taken as the nonmonotonic function

$$f(u) = \begin{cases} 1.14 + 0.3(u - 0.7) & u \geq 0.7 \\ 0.94 + (u - 0.5) & 0.5 \leq u \leq 0.7 \\ 0.84 + 0.5(u - 0.3) & 0.3 \leq u \leq 0.5 \\ 0.24 + (u + 0.3) & -0.3 \leq u \leq 0.3 \\ 0.18 + 0.2(u + 0.6) & -0.6 \leq u \leq -0.3 \\ 0.255 - 0.3(u + 0.85) & -0.85 \leq u \leq -0.6 \\ 0.43 - 0.5(u + 1.2) & y \leq -0.85 \end{cases} \tag{5.50}$$

The continuous time system was then sampled with a sampling interval of 0.08 s. Figure 5.4 shows the Bode plot of the linear block in the generating system, and the static nonlinear block.

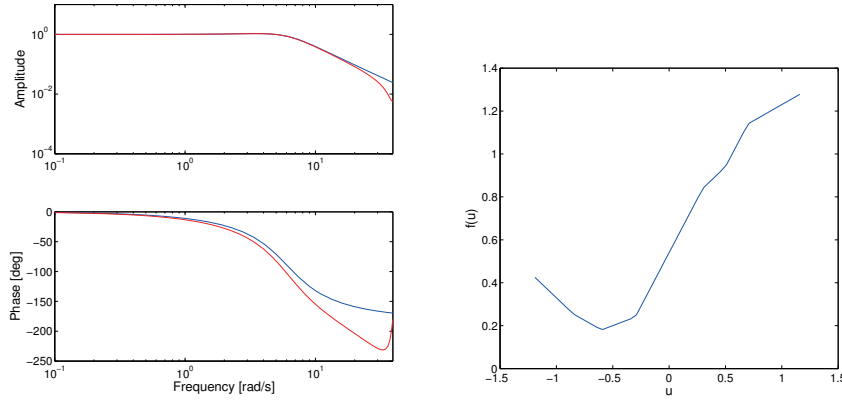


Figure 5.4: The generating system in Section 5.6.2. *Left:* The linear block of the generating system. Bode plot for the continuous time (blue, solid) and discrete time (red, dashed) model. *Right:* The static nonlinear block.

In order to highlight the effect due to errors in the estimated parameters, measurement noise with standard deviation of only 0.001 was added to the output.

The input signal for the example was chosen to be uniform white noise on the interval $(-1.2, 1.2)$. For the algorithm, the grid points where chosen to coincide with those of the system, i.e.,

$$\text{grid} = \begin{bmatrix} -1.2 & -0.85 & -0.6 & -0.3 & 0.3 & 0.5 & 0.7 & 1.2 \end{bmatrix}.$$

The slope of interval $I_o = [-0.3, 0.3]$ was chosen to be fixed equal to $k_o = 1$.

The algorithm was initialized with

$$\begin{aligned} \hat{\theta}(0) &= 0, \\ P(0) &= I, \\ \lambda(0) &= 0.95, \end{aligned}$$

and the user parameter λ_o was set equal to 0.99. After 3000 samples, the following parameter estimates where obtained

$$\begin{aligned} \hat{\theta}_l &= \begin{bmatrix} -1.392 & 0.570 & 0.097 & 0.081 \end{bmatrix}^\top, \\ \hat{\theta}_n &= \begin{bmatrix} 0.43 & -0.50 & -0.30 & 0.21 & 0.51 & 1.00 & 0.30 \end{bmatrix}^\top, \end{aligned}$$

while the true parameter vectors are

$$\begin{aligned} \theta_l^o &= \begin{bmatrix} -1.393 & 0.571 & 0.097 & 0.081 \end{bmatrix}^\top, \\ \theta_n^o &= \begin{bmatrix} 0.43 & -0.50 & -0.30 & 0.20 & 0.50 & 1.0 & 0.3 \end{bmatrix}^\top. \end{aligned}$$

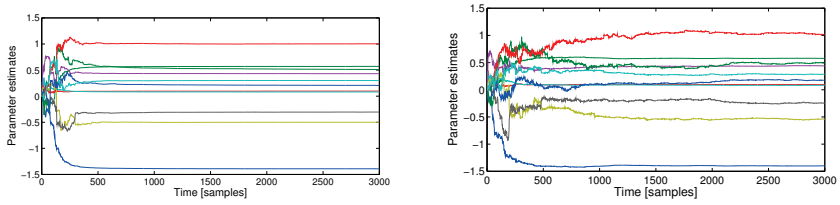


Figure 5.5: The parameter estimates in Section 5.6.2. *Left:* Measurement noise with standard deviation 0.001. *Right:* Measurement noise with standard deviation 0.05.

It can be concluded that the algorithm works as specified and that it converges towards the true parameter vector.

In order to see how measurement noise affects the performance of the algorithm, the same example was tested with measurement noise with standard deviation 0.05. After 3000 samples, the following parameter estimates where obtained

$$\hat{\theta}_l = \begin{bmatrix} -1.399 & 0.576 & 0.096 & 0.082 \end{bmatrix}^\top,$$
$$\hat{\theta}_n = \begin{bmatrix} 0.44 & -0.54 & -0.25 & 0.17 & 0.49 & 1.01 & 0.29 \end{bmatrix}^\top.$$

It can be concluded that the algorithm still converges towards the true parameter vector. In Figure 5.5 the evolution of the parameter estimates for both noise levels are shown.

5.6.3 A model of testosterone dynamics

Part II of this thesis concerns modelling of testosterone (Te) regulation. In this dynamical system testosterone secretion is stimulated by a hormone called luteinizing hormone (LH). A model for this is presented in Chapter 9, and if we add a saturation on the secretion of Te, then we get

$$y(t) = \frac{1}{s+b} f(x(t)), \tag{5.51}$$

where $y(t)$ is the Te concentration and x is a time average of the LH concentration. Here

$$f(x) = k_1 + k_2 \frac{(x/h)^p}{1 + (x/h)^p}. \tag{5.52}$$

As in other biomedical applications, a challenge is that the collected data sets are usually small. To counter that, models with few parameters have proved to provide a solution [88]. The algorithm of this paper was developed with this fact as a motivation.

Notice that the nonlinearity in (5.52) is not piecewise linear, and therefore the system described by (5.51)-(5.52) is *not* in the model structure we use for identification.

In order to demonstrate the behavior of Algorithm 2 when only few data samples are available, the system in (5.51) was sampled every 10 min, for 2000 minutes. Thus 200 samples was used in the identification. The input signal was generated as uniform white noise.

The parameters in (5.51)-(5.52) were set to $b = 0.046\,\mathrm{min}^{-1}$, $k_1 = 1$, $k_2 = 9$, $p = 4$ and $h = 4$. This value of b corresponds to the actual half-life of testosterone, cf. [79].

For the identification, the parameters where chosen as

$$\begin{aligned} k_0 &= 1 \\ \text{grid} &= \begin{bmatrix} 0 & 2 & 6 & 10 \end{bmatrix} \\ \hat{\theta}_l(0) &= 0 \\ \hat{\theta}_n(0) &= \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^\top \\ P(0) &= 10I \\ \lambda(0) &= 0.95 \\ \lambda_o &= 0.99 \end{aligned}$$

Notice that the choice of k_0 and grid means that the initial guess for the nonlinearity is a straight line with slope 1.

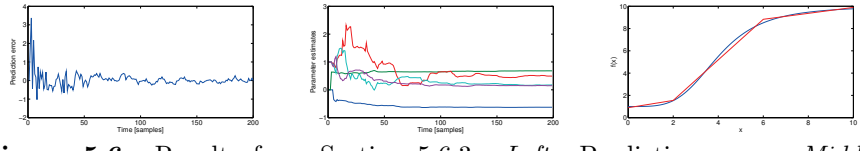


Figure 5.6: Results from Section 5.6.3. *Left:* Prediction error. *Middle:* Parameter estimates. *Right:* Scaled version of the estimated nonlinearity (red, dashed) and the true nonlinearity (solid, blue). The estimated nonlinearity is scaled, since the estimated and true system does not have the same static gain in the linear block.

The result of the identification gave a discrete time pole in 0.6262, which corresponds to $0.0453\,\mathrm{min}^{-1}$ in continuous time. The resulting prediction error and parameter estimates are shown in Figure 5.6, together with a scaled version of the estimated nonlinearity. The plots indicate that an accurate model could be estimated using only about 200 samples. However, this still assume well excited input, which is typically not the case in the real-world system.

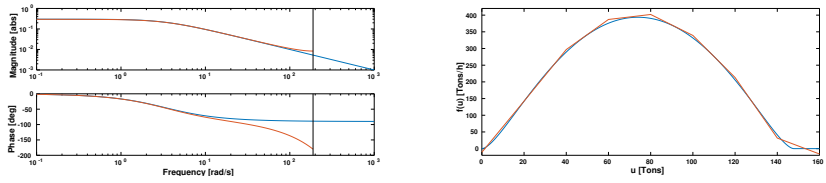


Figure 5.7: The true system in Section 5.6.4 (blue, solid) and the estimated (red, dashed).

5.6.4 A cement mill classifier

In this section, the algorithm is validated on a model of a cement mill classifier. The main purpose is to test the algorithm in a non-ideal situation, with a system that is not in the model structure. The classifier is part of the cement milling circuit, and the purpose is to separate the material into tailings (refused part), which are fed back into the mill, and the finished product. In [96], this part of the cement mill is modeled as a first order Hammerstein system,

$$T_f \dot{y} = -y + f(u), \tag{5.53}$$

where $T_f = 0.3\,\mathrm{h}$, y is the tailings flow rate (Tons/h) and u is the load on the mill (Tons). The static nonlinearity f is given by

$$f(u) = \frac{\phi(u)^{0.8} v^4}{K_\alpha + \phi(u)^{0.8} v^4}, \tag{5.54}$$

$$\phi(u) = \max\{0; -K_{\phi_1} u^2 + K_{\phi_2} u\}, \tag{5.55}$$

where v is the classifier speed (rpm). The constants are given by $K_\alpha = 3.57 \times 10^{10}\,(\mathrm{Tons/h})^{0.8}\mathrm{rpm}^4$, $K_{\phi_1} = 0.1116\,(\mathrm{Tons} \times \mathrm{h})^{-1}$, $K_{\phi_2} = 16.5\,\mathrm{h}^{-1}$. For illustration purposes, the model (5.53) was simulated with uniform white noise as input and sampled with a sampling period of 1 minute. The classifier speed was 140 rpm. Measurement noise with a standard deviation 1 was then added to the output.

The user parameters of the algorithm where chosen as $\lambda_o = 0.999$, $\lambda(0) = 0.95$, $P(0) = I$ and $\hat{\theta}(0) = 0$. For the identification, the nonlinearity was parameterized as in Section 5.5.2, with the grid points chosen as

$$\mathrm{grid} = \begin{bmatrix} 0 & 20 & 40 & 60 & 80 & 100 & 120 & 140 & 160 \end{bmatrix}.$$

Note that this means that the true nonlinearity is not in the model structure. The fixed parameter was chosen as b_1. After 5000 samples, the algorithm had estimated a discrete time pole in 0.945, which corresponds to a continues time pole in -3.36. The Bode plot and static nonlinearity of the true system and the estimated system are shown in

Figure 5.7. It can be seen that the algorithm converges toward a value close to the true pole, and it captures the static nonlinearity well even though it is not monotone and does not belong to the model structure.

5.A Implementation of the method

In order to implement the RPEM, we first need a way to compute the running estimates $\hat{y}(t)$ and $\psi(t)$. Here they are given by the proof of Lemma 5.1, in Appendix 5.B.1.

That is, let

$$\begin{aligned}\varphi(t+1) = [&-\hat{y}(t) \quad \cdots \quad -\hat{y}(t-n_a+1) \\ &f(u(t),\hat{\theta}(t)) \quad \cdots \quad f(u(t-n_b+1),\hat{\theta}_n(t))]^\top,\end{aligned} \tag{5.56}$$

then $\hat{y}(t+1) = \hat{\theta}_l^\top(t)\varphi(t+1)$.

Next, $\overline{\psi}_n(t+1) = \varphi_n(t+1)\hat{\theta}_l(t)$, where

$$\begin{aligned}\varphi_n(t+1) = [&-\varphi_n(t) \quad \cdots \quad -\varphi_n(t-n_a+1) \\ &F(u(t)) \quad \cdots \quad F(u(t-n_b+1))].\end{aligned} \tag{5.57}$$

For $\overline{\psi}_l(t)$, we note that the running estimates of $u_F(t,\theta)$ and $\hat{y}_F(t|\theta)$ are given by

$$u_F(t) = \hat{\theta}_a^\top(t)\begin{bmatrix} -u_F(t-1) \\ \vdots \\ -u_F(t-n_a)\end{bmatrix} + f(u(t),\hat{\theta}_a(t), \tag{5.58}$$

$$\hat{y}_F(t) = \hat{\theta}_a^\top(t)\begin{bmatrix} -\hat{y}(t-1) \\ \vdots \\ -\hat{y}_F(t-n_a)\end{bmatrix} + \hat{y}(t), \tag{5.59}$$

and

$$\begin{aligned}\psi_l(t+1) = [&-\hat{y}_F(t) \quad \cdots \quad -\hat{y}_F(t-n_a+1) \\ &-u_F(t) \quad \cdots \quad -u_F(t-n_a+1)]^\top.\end{aligned} \tag{5.60}$$

Using these running estimate in Algorithm 2, and replacing the R-recursion with the P-recursion in (5.22)-(5.24), we get Algorithm 5.

5.A.1 User parameters

The user parameters in Algorithm 5 are the initial values $\hat{\theta}(0)$, $P(0)$, and $\lambda(0)$, as well as the forgetting factor gain λ_o. For the forgetting

Algorithm 5 : RPEM for Hammerstein models

1: $\varepsilon(t) = y(t) - \hat{y}(t)$.
2: $\lambda(t) = \lambda_o \lambda(t-1) + 1 - \lambda_o$.
3: $S(t) = \psi^\top(t) P(t-1) \psi(t) + \lambda(t)$.
4: $P(t) = \frac{1}{\lambda(t)} \big(P(t-1) - P(t-1)\psi(t)\psi^\top(t)P(t-1)/S(t) \big)$.
5: $\hat{\theta}(t) = \Big[\hat{\theta}(t-1) + P(t)\psi(t)\varepsilon(t) \Big]_{\mathcal{D}_\mathcal{M}}$.
6: $\hat{y}(t+1) = \hat{\theta}_l^\top \varphi(t+1)$, with $\varphi(t+1)$ given by (5.56)
7: Compute $u_F(t)$ and $\hat{y}_F(t)$ as in (5.58)-(5.59).
8: $\psi_n(t+1) = \varphi_n(t+1)\hat{\theta}_l(t)$, with $\varphi_n(t+1)$ given by (5.57).
9: $\psi(t+1) = \begin{bmatrix} \psi_n(t+1) \\ \psi_l(t+1) \end{bmatrix}$ with $\psi_l(t+1)$ given by (5.60).

factor, a common choice is, as discussed in Section 5.3.1, $\lambda_o = 0.99$ and $\lambda(0) = 0.95$.

For P a common initial value is $P(0) = cI$, for some $c \in \mathbb{R}^+$, cf. Section 2.5.2. Usually c is chosen larger if the initial value $\hat{\theta}(0)$ is uncertain.

Finally, in order to keep the estimate within $\mathcal{D}_\mathcal{M}$, we need a projection algorithm. In the numerical examples of this chapter, we used

$$\Big[\hat{\theta}(t)\Big]_{\mathcal{D}_\mathcal{M}} = \begin{cases} \hat{\theta}(t) & \text{if } \hat{\theta}(t) \in \mathcal{D}_\mathcal{M} \\ \hat{\theta}(t-1) & \text{otherwise} \end{cases} . \tag{5.61}$$

5.B Proofs

5.B.1 Proof of Lemma 5.1

An easy way to see this is to note that, if $\theta \in \mathcal{D}_s$, then $\hat{y}(t|\theta)$ in (5.1), $\overline{\psi}_\ell(t,\theta)$ in (5.18) and $\overline{\psi}_n(t,\theta)$ in (5.19) are all given by stable linear filters applied to $F(u(t))$ or $\varphi(t,\theta)$. However, in this appendix, a constructive proof is given, since the constructions here are needed in order to implement the recursive algorithm.

For the following, it is useful to introduce the shift-matrix

$$S_n \triangleq \begin{bmatrix} 0 & 0 \\ I_{n-1} & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}.$$

Computing $\hat{y}(t|\theta)$.
First let

$$\xi_I(t,\theta) \triangleq \varphi(t,\theta).$$

It then follows from (5.11) that

$$\hat{y}(t|\theta) = \theta_\ell^\top \xi_I(t,\theta) = \begin{bmatrix} \theta_a^\top & \theta_b^\top \end{bmatrix} \xi_I(t,\theta) \triangleq C_1(\theta)\xi_I(t,\theta). \tag{5.62}$$

Now, using $f(u(t), \theta_n) = \theta_n^\top F(u(t))$, it follows that

$$\begin{aligned}\xi_I(t+1,\theta) &= \left(\begin{bmatrix} S_{n_a} & 0 \\ 0 & S_{n_b} \end{bmatrix} + \begin{bmatrix} -\theta_a^\top & -\theta_b^\top \\ 0 & 0 \end{bmatrix}\right) \xi_I(t,\theta) + \begin{bmatrix} 0 \\ \theta_n^\top \\ 0 \end{bmatrix} F(u(t)) \\ &\triangleq A_{11}(\theta)\xi_I(t,\theta) + B_1(\theta)F(u(t)).\end{aligned}$$

Note that $A_{11}(\theta)$ is block-triangular, so the eigenvalues are given by those of the diagonal blocks. The upper diagonal block is

$$S_{n_a} + \begin{bmatrix} -\theta_a^\top \\ 0 \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{n_a-1} & -a_{n_a} \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \tag{5.63}$$

for which the eigenvalues are the solutions to $\lambda^{n_a} + a_1\lambda^{n_a-1} + \cdots + a_{n_a-1}\lambda + a_{n_a} = 0$. Clearly, all these eigenvalues are inside the unit circle if $\theta \in \mathcal{D}_s$. Furthermore, the lower diagonal block is S_{n_b}, which has n_b eigenvalues in 0. Hence $A_{11}(\theta)$ has all eigenvalues inside the unit circle when $\theta \in \mathcal{D}_s$.

Computing $\overline{\psi}_n(t,\theta)$.
From (5.19), it follows that

$$\overline{\psi}_n(t,\theta) = \varphi_n(t)\theta_l, \tag{5.64}$$

where

$$\begin{aligned}\varphi_n(t,\theta) \triangleq [&-\overline{\psi}_n(t-1,\theta) \quad \cdots \quad -\overline{\psi}_n(t-n_a,\theta) \\ &F(u(t-1)) \quad \cdots \quad F(u(t-n_b))].\end{aligned} \tag{5.65}$$

Hence

$$\begin{aligned}\varphi_n(t+1,\theta) &= \varphi_n(t,\theta)\begin{bmatrix} S_{n_a}^\top & 0 \\ 0 & S_{n_b}^\top \end{bmatrix} + \begin{bmatrix} -\overline{\psi}_n(t,\theta) & 0 \end{bmatrix} + \begin{bmatrix} 0 & F(u(t)) & 0 \end{bmatrix} \\ &= \varphi_n(t,\theta)\left(\begin{bmatrix} S_{n_a}^\top & 0 \\ 0 & S_{n_b}^\top \end{bmatrix} + \begin{bmatrix} -\theta_l & 0 \end{bmatrix}\right) + \begin{bmatrix} 0 & F(u(t)) & 0 \end{bmatrix} \\ &= \varphi_n(t,\theta)A_{11}^\top(\theta) + \begin{bmatrix} 0 & F(u(t)) & 0 \end{bmatrix}. \end{aligned} \tag{5.66}$$

Now let

$$\xi_{II}(t,\theta) \triangleq \operatorname{vec}(\varphi_n(t,\theta)). \tag{5.67}$$

Using vectorization on (5.66), we get

$$\begin{aligned}\xi_{II}(t+1,\theta) &= (A_{11}(\theta) \otimes I)\,\xi_{II}(t,\theta) + \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix} F(u(t)) \\ &\triangleq A_{22}(\theta)\xi_{II}(t,\theta) + B_2(\theta)F(u(t)).\end{aligned}$$

Note that all eigenvalues of $A_{22}(\theta)$ are also eigenvalues of $A_{11}(\theta)$, so $A_{22}(\theta)$ has all eigenvalues inside the unit circle when $\theta \in \mathcal{D}_s$. Finally, it follows from (5.64) and (5.67) that

$$\overline{\psi}_n(t,\theta) = \left(\theta_l^\top \otimes I\right) \xi_{II}(t,\theta) \triangleq C_2(\theta)\xi_{II}(t,\theta).$$

Computing $\overline{\psi}_\ell(t,\theta)$.
From (5.18), it can be seen that $\overline{\psi}_\ell(t,\theta)$ is a filtered version of $\varphi(t,\theta)$. In order to compute this, introduce the filtered signals

$$u_F(t,\theta) \triangleq \frac{1}{A(q)} f(u(t),\theta_n), \quad \hat{y}_F(t,\theta) \triangleq \frac{1}{A(q)} \hat{y}(t|\theta). \tag{5.68}$$

Now let

$$\xi_{III}(t,\theta) \triangleq \begin{bmatrix} -\hat{y}_F(t-1,\theta) \\ \vdots \\ -\hat{y}_F(t-n_a,\theta) \end{bmatrix}, \quad \xi_{IV}(t,\theta) = \begin{bmatrix} -u_F(t-1,\theta) \\ \vdots \\ -u_F(t-n_a,\theta) \end{bmatrix}.$$

It then follows that

$$\overline{\psi}_\ell(t,\theta) = \begin{bmatrix} \xi_{III}(t,\theta) \\ \xi_{IV}(t,\theta) \end{bmatrix}.$$

Furthermore,

$$\hat{y}_F(t,\theta) = \theta_a^\top \xi_{III}(t,\theta) + \hat{y}(t|\theta) = \theta_a^\top \xi_{III}(t,\theta) + \theta_\ell^\top \xi_I(t,\theta),$$

so

$$\begin{aligned} \xi_{III}(t+1,\theta) &= \left(S_{n_a} + \begin{bmatrix} -\theta_a^\top \\ 0 \end{bmatrix} \right) \xi_{II}(t,\theta) + \begin{bmatrix} -\theta_\ell^\top \\ 0 \end{bmatrix} \xi_I(t,\theta) \\ &\triangleq A_{33}(\theta)\xi_{II}(t,\theta) + A_{31}(\theta)\xi_I(t,\theta). \end{aligned}$$

Note that $A_{33}(\theta)$ is given in (5.63), and it thus follows that $A_{33}(\theta)$ has all eigenvalues inside the unit circle when $\theta \in \mathcal{D}_s$.

In the same way, we can see that

$$u_F(t,\theta) = \theta_a^\top \xi_{IV}(t,\theta) + f(u(t),\theta_n) = \theta_a^\top \xi_{IV}(t,\theta) + \theta_n^\top F(u(t)),$$

so

$$\begin{aligned} \xi_{IV}(t+1,\theta) &= \left(S_{n_a} + \begin{bmatrix} -\theta_a^\top \\ 0 \end{bmatrix} \right) \xi_{IV}(t,\theta) + \begin{bmatrix} -\theta_n^\top \\ 0 \end{bmatrix} F(u(t)) \\ &\triangleq A_{44}(\theta)\xi_{IV}(t,\theta) + B_4(\theta)F(u(t)), \end{aligned}$$

where $A_{44}(\theta) = A_{33}(\theta)$, so $A_{44}(\theta)$ has all eigenvalues inside the unit circle when $\theta \in \mathcal{D}_s$.

Putting it all together
Now let

$$\xi(t,\theta) = \begin{bmatrix} \xi_I^\top(t,\theta) & \xi_{II}^\top(t,\theta) & \xi_{III}^\top(t,\theta) & \xi_{IV}^\top(t,\theta) \end{bmatrix}^\top.$$

From the above it can be seen that

$$\xi(t+1,\theta) = A(\theta)\xi(t,\theta) + B(\theta)F(u(t)),$$

where

$$A(\theta) = \begin{bmatrix} A_{11}(\theta) & 0 & 0 & 0 \\ 0 & A_{22}(\theta) & 0 & 0 \\ A_{31}(\theta) & 0 & A_{33}(\theta) & 0 \\ 0 & 0 & 0 & A_{44}(\theta) \end{bmatrix}, \quad B(\theta) = \begin{bmatrix} B_1(\theta) \\ B_2(\theta) \\ 0 \\ B_4(\theta) \end{bmatrix}.$$

Since $A_{11}(\theta), A_{22}(\theta), A_{33}(\theta)$ and $A_{44}(\theta)$ have all eigenvalues inside the unit circle when $\theta \in \mathcal{D}_s$, this is also holds for $A(\theta)$.

For the output, it follows from (5.62) that,

$$\hat{y}(t|\theta) = \begin{bmatrix} C_1(\theta) & 0 & 0 & 0 \end{bmatrix} \xi(t,\theta) \triangleq C_I(t,\theta)\xi(t,\theta),$$

and from (5.17) we get,

$$\psi(t,\theta) = I_o \begin{bmatrix} \overline{\psi}_n(t,\theta) \\ \overline{\psi}_\ell(t,\theta) \end{bmatrix} = I_o \begin{bmatrix} 0 & C_2(\theta) & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \xi(t,\theta) \triangleq C_{II}(t,\theta)\xi(t,\theta),$$

so

$$\begin{bmatrix} \hat{y}(t,\theta) \\ \psi(t,\theta) \end{bmatrix} = \begin{bmatrix} C_I(\theta) \\ C_{II}(\theta) \end{bmatrix} \xi(t,\theta) \triangleq C(\theta)\xi(t,\theta).$$

5.B.2 Proof of Lemma 5.2

Conditions M1 and M2 follow from Lemma 5.1 and the corresponding proof. Condition G1 and R1 follow from C5-C6.

To prove A3, first note that C1 implies that both $F(u(t))$ and $f(u(t),\theta_n^o)$ are strictly stationary since a static transformation does not affect strict stationarity. Since $\bar{y}(t)$ is generated by filtering $f(u(t),\theta_n^o)$ through an asymptotically stable linear system, it follows that $\bar{y}(t)$ tends to a strictly stationary stochastic process when time tends to infinity. Since $w(t)$ is strictly stationary, it holds that $y(t)$, and thus the data vector $z(t)$, are strictly stationary when time tends to infinity. Finally, the data $z(t)$ are fed into the asymptotically stable recursions given by (5.20)-(5.21). The reasoning above can then be repeated for $\varepsilon(t,\theta)$ and $\psi(t,\theta)$. This shows that both approach strict stationarity when time tends to infinity. Hence the limit in A3 exists.

Finally S1 has to be checked. This condition states that the data generation has to be exponentially stable. First note that (5.33) can be described as the state space model

$$\tilde{x}(t+1) = F(\theta^o)\tilde{x}(t) + G(\theta^o)f(u(t), \theta_n^o), \tag{5.69}$$
$$y(t) = H(\theta^o)\tilde{x}(t) + w(t). \tag{5.70}$$

As in, e.g., [162], let

$$z_s^o(t) = \begin{bmatrix} y_s^o(t) \\ F(u(t)) \end{bmatrix},$$

where $y_s^o(t)$ is the output from (5.69)-(5.70) initiated at time s with $\tilde{x}(s) = 0$ and where also the noise generation is initiated at time s. The state vector generated in this way is denoted $\tilde{x}_s^o(t)$. Clearly $\tilde{x}_s^o(t)$ and $y_s^o(t)$ are independent of anything that happened up to time s. It follows that

$$\begin{aligned} \|z(t) - z_s^o(t)\| &\leq \|y(t) - y_s^o(t)\| + \|w(t) - w_s^o(t)\| \\ &\leq C_1 \|\tilde{x}(t) - \tilde{x}_s^o(t)\| + \|w(t) - w_s^o(t)\| \end{aligned}$$

for some constant $C_1 < \infty$. Next note that,

$$\tilde{x}(t) - \tilde{x}_s^o(t) = \sum_{k=0}^{s-1} F^{t-k-1}(\theta^o)G(\theta^o)f(u(k), \theta_n^o).$$

Also note that $\|F^t(\theta^o)G(\theta^o)\| \leq C_2\lambda_1^t$ for some $C_2 < \infty$, $|\lambda| < 1$ since $\theta^o \in \mathcal{D}_\mathcal{M}$. Therefor, using the Cauchy-Schwartz inequality, it follows that

$$\begin{aligned} &\mathrm{E}\,\|\tilde{x}(t) - \tilde{x}_s^o(t)\|^4 \\ &\leq \mathrm{E}\left(\sum_{k=0}^{s-1} C_2^2\lambda_1^{2(t-k-1)}\right)^2 \left(\sum_{k=1}^{s-1} |f(u(k), \theta_n^o)|^2\right)^2 \\ &\leq C_3\left(\sum_{k=0}^{s-1} \lambda_1^{2(t-k-1)}\right)^2, \end{aligned}$$

where the last inequality follows from C3. From this and condition C4 condition S1 follow, cf. [160] page 44.

5.B.3 Proof of Lemma 5.3

An expression for $\psi(t,\theta)$ is given in (5.17). First consider $\overline{\psi}(t,\theta)$. Note that

$$\begin{aligned} A^2(q)\overline{\psi}_n(t,\theta) &= A(q)B(q)F(u(t)) = \\ &h_1F(u(t-1)) + \cdots + h_{n_a+n_b}F(u(t-n_a-n_b)) = \\ &\begin{bmatrix} h_1 I_{n_k} & \cdots & h_{n_a+n_b} I_{n_k} \end{bmatrix} \mathbf{F}_{n_a+n_b}(t) = (h^\top \otimes I_{n_k})\mathbf{F}_{n_a+n_b}(t) \end{aligned}$$

which implies that

$$\overline{\psi}_n(t,\theta) = (h^\top \otimes I_{n_k})v_{n_a+n_b}(t).$$

For $\overline{\psi}_\ell(t,\theta)$, define $u_F(t,\theta)$ and $\hat{y}_F(t,\theta)$ as in (5.68). Then

$$\begin{aligned} A^2(q)\hat{y}_F(t,\theta) &= b_1\bar{\theta}_n^\top F(u(t-1)) + \cdots + b_{n_b}\bar{\theta}_n^\top F(u(t-n_b)) \\ &= \begin{bmatrix} b_1\bar{\theta}_n^\top & \cdots & b_{n_b}\bar{\theta}_n^\top \end{bmatrix} \begin{bmatrix} F(u(t-1)) \\ \vdots \\ F(u(t-n_b)) \end{bmatrix} \\ &= \left(\begin{bmatrix} b_1 & \cdots & b_{n_b} \end{bmatrix} \otimes \bar{\theta}_n^\top\right) \mathbf{F}_{n_b}(t), \end{aligned}$$

and in the same way,

$$A^2(q)u_F(t,\theta) = \left(\begin{bmatrix} 1 & a_1 & \cdots & a_{n_a} \end{bmatrix} \otimes \bar{\theta}_n^\top\right) \mathbf{F}_{n_a+1}(t+1).$$

Hence,

$$A^2(q)\overline{\psi}_\ell(t,\theta) = \left(S(-B,A) \otimes \bar{\theta}_n^\top\right) \mathbf{F}_{n_a+n_b}(t),$$

so that

$$\overline{\psi}_\ell(t,\theta) = \left(S(-B,A) \otimes \bar{\theta}_n^\top\right) v_{n_a+n_b}(t),$$

and the lemma follows.

5.B.4 Proof of Lemma 5.4

First an expression for $\varepsilon(t,\theta)$ is derived. It follows from (5.32)-(5.33) that

$$y(t) = \bar{\theta}_l^{o\top}\varphi_o(t) + w(t)$$

where

$$\varphi_o(t) \triangleq \begin{bmatrix} -\bar{y}(t-1) & \cdots & -\bar{y}(t-n_a^o) & f(u(t-1),\theta_n^o) & \cdots & f(u(t-n_b^o),\theta_n^o) \end{bmatrix}^\top.$$

This implies

$$\begin{aligned} \varepsilon(t,\theta) &= \bar{\theta}_\ell^{o\top}\varphi_o(t) - \bar{\theta}_\ell^\top\varphi(t,\theta) + w(t) \\ &= \bar{\theta}_\ell^{o\top}(\varphi_o(t) - \varphi(t,\theta)) + (\bar{\theta}_\ell^o - \bar{\theta}_\ell)^\top\varphi(t,\theta) + w(t). \end{aligned} \tag{5.71}$$

Note that

$$\bar{\theta}_\ell^{o\top}\left(\varphi_o(t)-\varphi(t,\theta)\right)= \\ -\sum_{i=1}^{n_a} a_i^o\left(\varepsilon(t-i,\theta)-w(t-i)\right)+\sum_{i=1}^{n_b} b_i^o(\bar{\theta}_n^o-\bar{\theta}_n)^\top F(u(t-i)).$$

Inserting this into (5.71) gives

$$A_o(q)\varepsilon(t,\theta)=B_o(q)(\bar{\theta}_n^o-\bar{\theta}_n)^\top F(u(t))+A_o(q)w(t)+(\bar{\theta}_\ell^o-\bar{\theta}_\ell)^\top\varphi(t,\theta)$$

and thus

$$\varepsilon(t,\theta)=\left(\bar{\theta}^o-\bar{\theta}\right)^\top\frac{1}{A_o(q)}\begin{bmatrix}B_o(q)F(u(t))\\ \varphi(t,\theta)\end{bmatrix}+w(t).$$

Next note that $\bar{\theta}^o-\bar{\theta}$ is assumed to be zero where the fixed parameter is located, so $\bar{\theta}^o-\bar{\theta}=I_o^\top(\theta^o-\theta)$, and it follows that

$$\begin{aligned}\varepsilon(t,\theta)&=(\theta^o-\theta)^\top\frac{1}{A_o(q)}I_o\begin{bmatrix}B_o(q)F(u(t))\\ \varphi(t,\theta)\end{bmatrix}+w(t)\\ &=(\theta^o-\theta)^\top\tilde{\psi}(t,\theta)+w(t).\end{aligned}$$

Hence

$$\begin{aligned}f_A(\theta)&=\lim_{t\to\infty}\mathrm{E}\,\psi(t,\theta)\varepsilon(t,\theta)\\ &=\lim_{t\to\infty}\left(\mathrm{E}\,\psi(t,\theta)\tilde{\psi}^\top(t,\theta)(\theta^o-\theta)+\mathrm{E}\,\psi(t,\theta)w(t)\right).\end{aligned}$$

Using C1-C2 it can be seen that $\lim_{t\to\infty}\mathrm{E}\,\psi(t,\theta)w(t)=0$, and thus the lemma follows.

5.B.5 Proof of Lemma 5.6

First assume that $M(\theta^o)PM^\top(\theta^o)\succ 0$. Choose any α such that $\alpha^\top G_A(\theta^o)\alpha=0$. From (5.28) and Lemma 5.3 it follows that

$$\lim_{t\to\infty}\alpha^\top\mathrm{E}\left[M(\theta^o)v_{n_a+n_b}(t,\theta^o)v_{n_a+n_b}^\top(t,\theta^o)M(\theta^o)^\top\right]\alpha=0,$$

and hence

$$\lim_{t\to\infty}\alpha^\top M(\theta^o)v_{n_a+n_b}(t,\theta^o)=0,\quad\text{w.p.1.}\tag{5.72}$$

It follows from (5.72) and (5.40) that, with probability one,

$$0=\lim_{t\to\infty}\alpha^\top M(\theta^o)\left(A_o^2(q)v_{n_a+n_b}(t,\theta^o)\right)=\lim_{t\to\infty}\alpha^\top M(\theta^o)\mathbf{F}_{n_a+n_b}(t).$$

Since $\mathbf{F}_{n_a+n_b}(t)$ is a strictly stationary process when condition C1 holds, the limit can be removed, so

$$\alpha^\top M(\theta^o)\mathbf{F}_{n_a+n_b}(t) = 0, \quad \text{w.p.1.}$$

Hence, using the definition in (5.43), it follows that

$$\alpha^\top M(\theta^o)PM^\top(\theta^o)\alpha = 0.$$

Since we assumed $M(\theta^o)PM^\top(\theta^o) \succ 0$, this implies $\alpha = 0$. We can thus conclude that the only α such that $\alpha^\top G_A(\theta^o)\alpha = 0$ is $\alpha = 0$, so $G_A(\theta^o) \succ 0$.

For sufficiency, assume instead that $G_A(\theta^o) \succ 0$, and choose any α such that $\alpha^\top M(\theta^o)PM^\top(\theta^o)\alpha = 0$. Then

$$\alpha^\top M(\theta^o)\mathbf{F}_{n_a+n_b}(t) = 0, \quad \text{w.p.1.}$$

Since $1/A_o^2(q)$ is an asymptotically stable filter it follows that

$$\lim_{t\to\infty} \alpha^\top M(\theta^o)v_{n_a+n_b}(t,\theta^o) = 0 \quad \text{w.p.1,}$$

and thus

$$\alpha^\top G(\theta^o)\alpha = 0,$$

so it can be concluded that $\alpha = 0$, and thus $M(\theta^o)PM^\top(\theta^o) \succ 0$.

5.B.6 Proof of Lemma 5.7

To see this, first note that $M(\theta)$ has dimension $(n_k + n_a + n_b - 1) \times (n_a + n_b)n_k$, and it thus has more columns than rows. Also note that condition C8 implies that the Sylvester matrix $S(-B, A)$ is non-singular [144]. Hence, $M(\theta)$ has full row rank if and only if $M(\theta)M^\top(\theta)$ is positive definite.

When the fixed parameter is k_i.
Let $\tilde{I}_o$ be I_{n_k} with row i removed. It then follows from (5.38) that

$$M(\theta) = \begin{bmatrix} h^\top \otimes \tilde{I}_o \\ S(-B, A) \otimes \bar{\theta}_n^\top \end{bmatrix}. \tag{5.73}$$

Noticing that $h^\top h$ and $\bar{\theta}_n^\top \bar{\theta}_n$ are scalars, that $\tilde{I}_o \tilde{I}_o^\top = I_{n_k-1}$, and that $\bar{\theta}_n^\top \tilde{I}_o^\top \tilde{I}_o \bar{\theta}_n = \bar{\theta}_n^\top \bar{\theta}_n - k_i^2$, it follows by straightforward multiplication and application of the determinant formula for block matrices that

$$\begin{aligned} &\det[M(\theta)M^\top(\theta)] = \\ &\qquad (h^\top h)^{n_k-1} \det\left[SS^\top(\bar{\theta}_n^\top \bar{\theta}_n) - \frac{Shh^\top S^\top}{h^\top h}(\bar{\theta}_n^\top \bar{\theta}_n - k_i^2)\right] = \\ &\qquad (h^\top h)^{n_k-1} \det(SS^\top) \det\left[I_{n_a+n_b}\bar{\theta}_n^\top \bar{\theta}_n - \frac{hh^\top}{h^\top h}(\bar{\theta}_n^\top \bar{\theta}_n - k_i^2)\right], \end{aligned} \tag{5.74}$$

where $S = S(-B, A)$. By using the formula $\det(I+AB) = \det(I+BA)$, it follows that

$$\det\left[I_{n_a+n_b}\bar{\theta}_n^\top\bar{\theta}_n - \frac{hh^\top}{h^\top h}(\bar{\theta}_n^\top\bar{\theta}_n - k_i^2)\right] = (\bar{\theta}_n^\top\bar{\theta}_n)^{n_a+n_b}\left(1 - \frac{\bar{\theta}_n^\top\bar{\theta}_n - k_i^2}{\bar{\theta}_n^\top\bar{\theta}_n}\right) = (\bar{\theta}_n^\top\bar{\theta}_n)^{n_a+n_b-1}k_i^2$$

and hence

$$\det(M(\theta)M^\top(\theta)) = (h^\top h)^{n_k-1}(\bar{\theta}_n^\top\bar{\theta}_n)^{n_a+n_b-1}k_i^2\det(SS^\top) \tag{5.75}$$

Since the Sylvester matrix S is non-singular, the above expression is non-zero, and it follows that $M(\theta)$ has full rank.

When the fixed parameter is b_i.
Let S_o be the matrix resulting from deleting row $n_a + i$ from $S(-B, A)$, that is, delete the row corresponding to b_i in $S(-B, A)$. Then, with b_i fixed, (5.38) reduces to

$$M(\theta) = \begin{bmatrix} h^\top \otimes I_{n_k} \\ S_o \otimes \bar{\theta}_n^\top \end{bmatrix}. \tag{5.76}$$

In the same way as in the previous case, it can be seen that

$$\det(M(\theta)M^\top(\theta)) = (h^\top h)^{n_k}\det\left[S_oS_o^\top(\bar{\theta}_n^\top\bar{\theta}_n) - \frac{S_ohh^\top S_o^\top}{h^\top h}(\bar{\theta}_n^\top\bar{\theta}_n)\right] = (h^\top h)^{n_k}(\bar{\theta}_n^\top\bar{\theta}_n)^{n_a+n_b-1}\det[S_oS_o^\top]\det\left[I - (S_oS_o^\top)^{-1}\frac{S_ohh^\top S_o^\top}{h^\top h}\right].$$

Note that

$$\det\left[I - (S_oS_o^\top)^{-1}\frac{S_ohh^\top S_o^\top}{h^\top h}\right] = \frac{1}{h^\top h}\left(h^\top h - h^\top S_o^\top(S_oS_o^\top)^{-1}S_oh\right) = \frac{1}{h^\top h}\left\|S_o^\top x - h\right\|_2^2,$$

where $x = (S_oS_o^\top)^{-1}S_oh$. Hence,

$$\det(M(\theta)M^\top(\theta)) = (h^\top h)^{n_k-1}(\bar{\theta}_n^\top\bar{\theta}_n)^{n_a+n_b-1}\det[S_oS_o^\top]\left\|S_o^\top x - h\right\|_2^2. \tag{5.77}$$

Therefore, it follows that $M(\theta)$ has full row rank if and only if there is no x such that $S_o^\top x = h$. Note that

$$h = S^\top(-B, A) \begin{bmatrix} \mathbf{0}_{n_a}^\top & b_1 & \cdots & b_{n_b} \end{bmatrix}^\top, \tag{5.78}$$

$$S_o^\top x - h = S^\top(-B, A) \left(\begin{bmatrix} x_1 \\ \vdots \\ x_{n_a+i-1} \\ 0 \\ x_{n_a+i} \\ \vdots \\ x_{n_a+n_b-1} \end{bmatrix} - \begin{bmatrix} \times \\ \vdots \\ \times \\ b_i \\ \times \\ \vdots \\ \times \end{bmatrix} \right), \tag{5.79}$$

where $\times$ denotes an arbitrary element. This expression can only be zero if $b_i = 0$, since $S^\top(-B, A)$ is non-singular. Since it is assumed that the fixed parameter b_i is non-zero, it follows that $M(\theta)$ has full rank.

Remark 5.6. *The expression in* (5.77) *also holds when the fixed parameter is a_i and S_o is $S(-B, A)$ with the row corresponding to a_i removed. In the same way as in* (5.78)-(5.79)*, it can be seen that, in this, case there is an x such that $S_o^\top x = h$, and thus $M(\theta)$ can not have full rank.*

5.B.7 Proof of Lemma 5.8

When C10 holds, this lemma follows trivially from Lemma 5.6 and Lemma 5.7. However, when C11 holds instead, P does not have full rank. To prove that the lemma still holds, it will be shown that, for any ρ such that $P\rho = 0$, there is no r such that $M(\theta^o)r = \rho$.

Therefore, assume that C11 holds and let ρ be such that $P\rho = 0$. Then

$$0 = \rho^\top \mathrm{E}(\mathbf{F}_{n_a+n_b}(t) - m_{\mathbf{F}})(\mathbf{F}_{n_a+n_b}(t) - m_{\mathbf{F}})^\top \rho,$$

where $m_{\mathbf{F}} = \mathrm{E}\,\mathbf{F}_{n_a+n_b}(t)$. With probability one it follows that

$$0 = \rho^\top(\mathbf{F}_{n_a+n_b}(t) - m_{\mathbf{F}}) = \sum_{i=1}^{n_a+n_b} \rho_i^\top(F(u(t-i)) - m_F),$$

where $m_F = \mathrm{E}\,F(u(t))$. Since $f_1(u) = 1$, this implies that

$$0 = \sum_{i=1}^{n_a+n_b} \rho_i^\top \begin{bmatrix} 0 \\ F_o(u(t-i)) - m_{F_o} \end{bmatrix}, \text{ w.p.1},$$

where $m_{F_o} = \mathrm{E}\,F_o(u(t))$. Given the assumption that $F_o(u)$ is persistently exciting of order $n_a + n_b$, it follows that there exists $p_i \in \mathbb{R}$ such

that $\rho_i^\top = \begin{bmatrix} p_i & 0 & \cdots & 0 \end{bmatrix}$ for all i. Hence

$$\left(I_{n_a+n_b} \otimes \begin{bmatrix} 0 & I_{n_k-1} \end{bmatrix}\right) \rho = 0. \tag{5.80}$$

Furthermore, if $P\rho = 0$, it also holds that

$$0 = \mathrm{E}\, \mathbf{F}_{n_a+n_b}(t)\mathbf{F}_{n_a+n_b}^\top(t)\rho = \mathrm{E}\, \mathbf{F}_{n_a+n_b}(t) \sum_{i=1}^{n_a+n_b} p_k,$$

since $\mathbf{F}_{n_a+n_b}(t)$ is one at all indices where ρ is nonzero. Also, note that $\mathrm{E}\, \mathbf{F}_{n_a+n_b}(t) \neq 0$, since $f_1(u) = 1$. Hence,

$$\sum_{k=1}^{n_a+n_b} p_k = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \rho = 0. \tag{5.81}$$

So, for any r that satisfies $M^\top(\theta)r = \rho$, it follows from (5.80) that

$$\begin{aligned} 0 &= (I_{n_a+n_b} \otimes \begin{bmatrix} 0 & I_{n_k-1} \end{bmatrix}) M^\top(\theta) r \\ &= \begin{bmatrix} 0 & (h \otimes I_{n_k-1}) & (S^\top(-B, A) \otimes \begin{bmatrix} 0 & I_{n_k-1} \end{bmatrix} \bar{\theta}_n) \end{bmatrix} I_o^\top r \\ &= \begin{bmatrix} 0 & \bar{M}^\top(\theta) \end{bmatrix} r. \end{aligned}$$

In the same way as in Lemma 5.7 it can now be shown that $\overline{M}(\theta^o)$ has full row-rank, since $\overline{M}(\theta^o)$ is defined in the same way as $M(\theta^o)$ but without the k_1 parameter. It can thus be concluded that $r = \begin{bmatrix} r_1 & 0 & \cdots & 0 \end{bmatrix}^\top$. Hence it follows from (5.81) that

$$0 = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} M^\top(\theta^o) \begin{bmatrix} r_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = r_1 A_o(1) B_o(1)$$

since the first row in $M(\theta^o)$ contains all the coefficients of the polynomial $A_o(q)B_o(q)$ and the rest of the entries are zero. Hence, it can be concluded that $r_1 = 0$, so $r = 0$ and thus $\rho = 0$. That is, the only ρ such that $P\rho = 0$ for which there exist a solution to $M^\top(\theta^o)r = \rho$, is $\rho = 0$. Hence it follows that $G(\theta^o)$ is positive definite.

5.B.8 Proof of Corollary 5.1

We prove this for the first case, and the second case can be showed in the exact same way. Hence, assume that $\mathrm{E}(F(u(t)) - m_F)(F(u(t)) - m_F)^\top$ is positive definite. The goal is then to show that

$$\mathrm{E}\, \mathbf{F}_{n_a+n_b}(t)\mathbf{F}_{n_a+n_b}(t)$$

is positive definite, but will in fact show the stronger statement:

$$\mathrm{E}(\mathbf{F}_{n_a+n_b}(t) - m_{\mathbf{F}})(\mathbf{F}_{n_a+n_b}(t) - m_{\mathbf{F}})^\top \succ 0.$$

To do this, let ρ be such that

$$\rho^\top \mathrm{E}(\mathbf{F}_{n_a+n_b}(t) - m_{\mathbf{F}})(\mathbf{F}_{n_a+n_b}(t) - m_{\mathbf{F}})^\top \rho = 0. \tag{5.82}$$

Note that, if this implies that $\rho = 0$, then C10 holds. To see that this is indeed the case, note that (5.82) implies

$$0 = \rho^\top(\mathbf{F}_{n_a+n_b}(t) - m_{\mathbf{F}}) = \sum_{i=1}^{n_a+n_b} \rho_i^\top (F(u(t-i)) - m_F) \quad \text{w.p.1.}$$

Multiply this expression from the right with $(F(t-k) - m_F)^\top$ and take the expectation to get

$$\mathrm{E}\left\{ \sum_{i=1}^{n_a+n_b} \rho_i^\top (F(u(t-i)) - m_F)(F(u(t-k)) - m_F)^\top \right\} = 0$$

By the assumption that $u(t)$ is white noise, and the fact that $F(u(t))$ is just a static function of $u(t)$, it follows that $F(u(t-i))$ and $F(u(t-k))$ are uncorrelated when $i \neq k$, and thus

$$\rho_k^\top \mathrm{E}(F(u(t-k) - m_F)(F(u(t-k)) - m_F)^\top = 0.$$

By the assumption that $\mathrm{E}(F(u(t-k)) - m_F)(F(u(t-k)) - m_F)^\top$ is positive definite, it follows that $\rho_k = 0$. This holds for all k, so we can conclude that $\rho = 0$, and thus C10 holds.

5.B.9 Proof of Lemma 5.10

From Corollary 5.1, it follows that it is enough to show that

$$\mathrm{E}[(F_o(u(t)) - m_{F_o})(F_o(u(t)) - m_{F_o})^\top] \succ 0. \tag{5.83}$$

Let α be such that

$$\alpha^\top \mathrm{E}[(F_o(u(t)) - m_{F_o})(F_o(u(t)) - m_{F_o})^\top]\alpha = 0.$$

If this implies that $\alpha = 0$, then (5.83) and thus C11 holds. To see that this is indeed the case, note that the above equation implies

$$0 = \alpha^\top (F_o(u(t)) - m_{F_o}), \quad \text{w.p.1.}$$

Choose any $k \in \{2, \ldots, n_k\}$. According to (5.47) there is a positive probability that $u(t) \in I_k$, and in this case

$$\alpha^\top (F_o(u(t)) - m_{F_o}) = \sum_{i=2}^{n_k} \alpha_{i-1} f_i(u(t)) - \sum_{i=2}^{n_k} \alpha_{i-1} m_{f_i} =$$
$$\alpha_{k-1}(u(t) - u_{k-1}) + \sum_{i=2}^{k-1} \alpha_{i-1}(u_i - u_{i-1}) - \sum_{i=2}^{n_k} \alpha_{i-1} m_{f_i},$$

where $m_{f_i} = \mathrm{E}[f_i(u)]$. Since only the first term depends on $u(t)$, and $p_u(u) \geq \delta > 0$ in at least one nonempty subinterval of I_k, we must have $\alpha_{k-1} = 0$. This holds for all $k \in \{2, \ldots, n_k\}$, so $\alpha = 0$.

Part II:
Modeling of testosterone regulation

Chapter 6

Modeling of endocrine systems

This chapter gives a brief introduction to endocrine systems, with a special focus on testosterone regulation in the human male.

6.1 Introduction

The research field of system biology has gained significant popularity during the last decades. System biology can be seen as the systematic study of complex interactions in biological system, mainly by methods from theory of dynamical system. A living organism is a complex dynamical system comprising numerous interacting subsystems equipped with multiple feedback and feedforward mechanisms. One of the reasons for the success of system biology is that it describes the biological function by means of multi-scale mathematical models that can be subjected to mathematical analysis within the framework of dynamical systems theory. Based on this solid ground, predictions about system behaviors arising in response to parametric changes and exogenous stimuli can be made and experimentally tested.

As discussed in Section 1.4, there are several different purposes of constructing mathematical models for dynamical systems. In biomedicine, the model can provide new insights into the biological system at hand. Further, these insights, together with suitable mathematical tools from e.g. control theory, can be used to construct new strategies for medical interventions.

6.2 The endocrine system

The endocrine system consists of all the cells, glands, and tissues that produce hormones in a living organism. Hormones are molecules that

are synthesized by the glands and secreted into the bloodstream. In this way, hormones act as chemical messengers that transfer information between cells, and thus, also organs.

The endocrine system also communicates with the nervous systems through the hypothalamus of the brain. The nervous system sends information to hypothalamus about changes in the body, and regulates the production of hormones in the pituitary gland. In this way, both the endocrine and nervous system use feedback to control physiological and behavioral processes of the organism. Due to feedback loops, the secretion of a hormone is stimulated (or inhibited) by other hormones. That is, the secretion rate of one hormone depends on the concentration (or concentration change rate) of other hormones. After hormone molecules have been synthesized and released into the bloodstream, they maintain a biologically active state for a while, and then, like all organic molecules, they degrade. The process of degradation is referred to as elimination or clearing of hormones.

The above description of the endocrine system, with secretion and elimination rates, as well as feedback loops, indicates that it can be seen as a dynamical system. This thus suggests that the theory of dynamical systems could be useful for modelling and analysis of endocrine systems.

6.2.1 Mathematical models of endocrine systems

As seen in Section 1.3, we can take a number of different approaches in constructing mathematical models of endocrine systems. One possibility is to derive a model from fundamental principles of biology, biochemistry, and physics. This approach has been followed in e.g. modeling of the glucose-insulin feedback system in diabetes 1 [82], and in simulating the mechanisms of the human menstrual cycle [125].

However, since endocrine systems usually consist of complex networks of interacting glands and hormones, such models are typically of high dimension and very cumbersome to develop. In order to obtain insight into the principles of biological feedback via mathematical analysis, a modelling approach where only the most essential characteristics and interactions of the system are included appears to be plausible. With this approach, it is also often easier to apply techniques from system identification in order to estimate the parameters of the model in a systematic way, yielding a relatively simple model that still can accurately describe how the hormone concentrations vary with time.

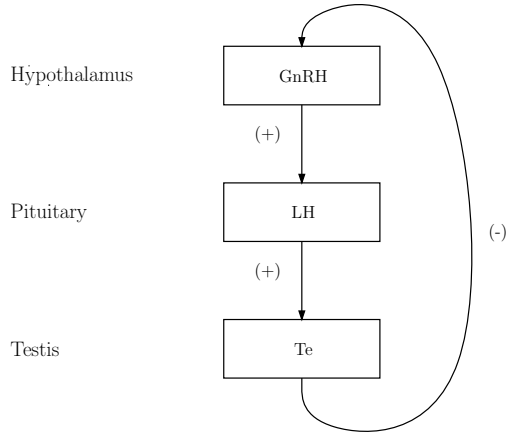


Figure 6.1: Schematic diagram of the male hypothalamic-pituitary-gonadal system. Arrows denote feedforward (stimulatory (+)) and feedback (inhibitory (-)) actions.

6.3 Testosterone regulation

The endocrine system can be divided into several subsystems that are responsible for different physiological functions. In mammals, one important objective of the endocrine system is to control the reproductive system. In the human male, this is primarily done through the regulation of the male sex hormone – testosterone (Te). Testosterone levels are also involved in the growth of muscle, bones and fat tissue, among other things.

The testosterone regulation system is schematically depicted in Figure 6.1. It mainly comprises two other hormones: luteinizing hormone (LH) and gonadotropin-releasing hormone (GnRH). GnRH is a neurohormone released in the hypothalamus of the brain. GnRH then stimulates the secretion of LH by the pituitary gland. LH travels through the bloodstream to the testes where it stimulates the secretion of Te. Finally, the feedback loop is closed by Te inhibiting the secretion of both LH and GnRH [153]. However, the inhibition of LH has a relatively small effect on the closed-loop system, and is therefore not considered in this thesis.

There are several reasons for studying the dynamics of testosterone regulation in the human male. To start with, a pragmatic reason is that it is one of the simplest endocrine subsystems, involving mainly three different hormones in the axis GnRH-LH-Te. At the same time, the regulation of Te shares many common aspects with regulation in other parts of the endocrine systems, e.g., those handling cortisol, growth hormone, and insulin. Thus, methods developed for modeling of Te-

regulation in this thesis can hopefully be applied to other parts of the endocrine system in the future.

Furthermore, Te-regulation, being an integral part of the reproductive system in the human male, constitutes an interesting research topic on its own right. Understanding how the Te-regulation works would help planning medical interventions such as testosterone replacement therapy. Accurate mathematical models can then help researchers to get a deeper insights into the closed-loop dynamics in Te-regulation, as well as be an aid in optimizing therapies.

6.3.1 The Smith model

In [133], a reductionist approach was taken to develop a qualitative mathematical model describing the male reproductive system. The model, usually called the Smith model, can be expressed by the following three ordinary differential equations

$$\begin{aligned} \dot{R} &= f(T) - B_1(R), \\ \dot{L} &= G_1(R) - B_2(L), \\ \dot{T} &= G_2(L) - B_3(T), \end{aligned} \tag{6.1}$$

where $R(t)$, $L(t)$, and $T(t)$ represent the serum concentration of GnRH, LH and Te, respectively. The non-negative functions B_1, B_2, B_3 describe the clearing rates of the hormones and G_1, G_2, f specify the rates of their secretion. Instead of portraying all the complex interactions between the constituting parts of the endocrine system, the Smith model captures only the main features of Figure 6.1.

It is well known that the concentrations of LH and Te do not stay at constant levels, so only self-sustained oscillating solutions are biologically feasible behaviors of the autonomous model in (6.1). The dynamical properties of (6.1) have been studied analytically to great extent for different choices of the functions B_i, G_i and f. It is relatively common to approximate B_i and G_i by linear functions, so that

$$\begin{aligned} B_i(x) &= b_i x, \quad b_i > 0, i = 1, 2, 3; \\ G_i(x) &= g_i x, \quad g_i > 0, i = 1, 2. \end{aligned} \tag{6.2}$$

In [133], sufficient conditions for (6.1)-(6.2) to have stable periodic solutions are given. Figure 6.2 shows two solutions of (6.1)-(6.2) with $f(x)$ chosen to be a Hill function,

$$f(x) = \frac{K}{1 + \beta x^p}. \tag{6.3}$$

It can be seen that the solution converges to a stable stationary point

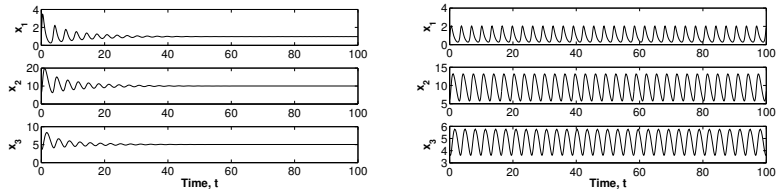


Figure 6.2: Solutions to the Smith model with the nonlinear function f as a Hill function. Left, Hill order $\rho = 7$. Right, Hill order $\rho = 10$.

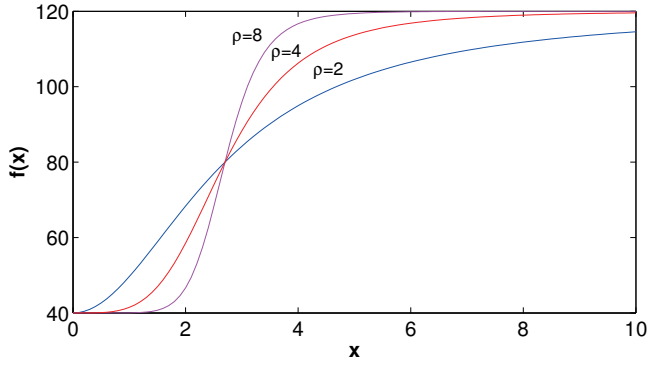


Figure 6.3: The Hill function (6.3) for different values of ρ.

when $\rho = 7$, and that the solution is periodic when $\rho = 10$ for the chosen parameter values. In fact, it was shown in [47] that a necessary condition for a periodic solution is that $\rho > 8$, for any choices of the linear parameters. In e.g. [69] and [64], it is argued that such a high value of the Hill function order is unrealistic. Indeed, already for $\rho \geq 4$ (cf. Figure 6.3), the Hill function in (6.3) resembles a relay characteristic that lacks a proper biological justification.

To resolve the above issue, several attempts have been made to extend the Smith model in such a way that the solutions oscillate for a broader range of the parameters, mostly by introducing special types of nonlinear feedback and time delays. Yet, the Smith model of testosterone regulation has been proven to be asymptotically stable for any value of the time delay under a nonlinear feedback in the form of a first-order Hill function, [43]. However, by introducing a non-smooth feedback such as piecewise affine nonlinearities, multiple periodical orbits and chaos arise in the Smith model [2, 139]. Multiple delays in the Smith model under a second-order Hill function feedback have also been shown to lead to sustained nonlinear oscillations in some subspaces of the model parameters [41].

6.3.2 Convolution models

As described in Section 6.3.1, there are several problems with using the classical closed-loop Smith model. For this reason, at present, analysis of hormone dynamics from measured (blood serum) hormone concentrations usually only considers open-loop dynamics. It is usually assumed that the concentration of a single hormone satisfies a linear ordinary differential equation of the form

$$\frac{dC(t)}{dt} = -bC(t) + S(t), \tag{6.4}$$

where $C(t)$ is the time profile of the concentration, $S(t)$ the hormone secretion rate, and b is the rate of elimination, see, e.g. [79]. So, if the hormone in question is LH, then $C(t)$ and $S(t)$ correspond respectively to $L(t)$ and $G_1(R(t))$ in the Smith model (6.1)-(6.2), while b corresponds to b_2.

For (6.4), it holds that

$$C(t) = \int_0^t S(\tau)\Phi_b(t-\tau)d\tau + C(0)E(t), \tag{6.5}$$

where $\Phi_b(t) = e^{-bt}$ is the impulse response of (6.4) and describes the elimination rate profile of the hormone. A reasonable and often used model for a hormone concentration is therefore the convolution integral given by (6.5). In order to better understand the properties of the endocrine system, it is of interest to obtain the time profile of the secretion rate $S(t)$, while only the concentration $C(t)$ is measurable. In this case, deconvolution methods can be used to estimate $S(t)$ from $C(t)$ by exploiting (6.4).

However, if there is no *a priori* information about the secretion profile $S(t)$ or the impulse response $\Phi_b(t)$, this problem is usually ill-conditioned. Hence, in most cases, some assumptions on the secretion profile are made, resulting in so-called model-based deconvolution. Many algorithms for estimation of hormone secretion rates from concentration data exist, e.g. WEN Deconvolution [35], WINSTODEC [138], and AutoDecon [71]. In [35], a comparison of several different deconvolution approaches is presented.

Most deconvolution-based methods capture major pulsatile secretion events when used on longer time series. However, they tend to neglect the existence of smaller pulses in between major pulses [65]. In [65] some methods for circumventing this problem are proposed. Further, the deconvolution-based methods do not take into account the fact that the hormones in the endocrine system are part of a closed-loop system.

6.3.3 The pulse-modulated Smith model

It is well known that GnRH is released episodically by hypothalamic neurons in modulated secretory bursts [38], and in e.g. [34], a detailed mathematical description of GnRH pulses is presented.

The pulsatile nature of GnRH is not directly reflected by the classical Smith model in (6.1), where the resulting oscillating temporal profile of the involved hormones is due to smooth nonlinearities.

However, for the purpose of capturing the biological feedback mechanism, an element implementing pulse-amplitude and pulse-frequency modulation can be used, as explained in [157]. Thus the mathematical framework of pulse-modulated feedback, discussed in Chapter 7, is directly applicable.

A version of the Smith model that makes use of pulse-modulation was introduced in [30].

In order to extend the Smith model with pulse-modulation, let t_k, $k = 0, 1, 2, \ldots$ be the time instances when a GnRH pulse is released, and let w_k represent the weight of impulse number k. Then a pulse-modulated model of testosterone regulation can be written in a state-space form with $x \in \mathbb{R}^3$, and $x_1 = R(t)$, $x_2(t) = L(t)$, $x_3(t) = T(t)$, as

$$\dot{x}(t) = \begin{bmatrix} -b_1 & 0 & 0 \\ g_1 & -b_2 & 0 \\ 0 & g_2 & -b_3 \end{bmatrix} x(t), \qquad \text{if } t \neq t_k \tag{6.6}$$

$$x(t_k^+) = x(t_k^-) + w_k \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad \text{if } t = t_k \tag{6.7}$$

where b_1, b_2, b_3, g_1 and g_2 are positive parameters as in (6.2). The impulse times t_k and the impulse weights w_k are given by the recursion

$$t_{k+1} = t_k + T_k, \quad T_k = \Phi(x_3(t)), \quad w_k = F(x_3(t)), \tag{6.8}$$

where $\Phi(\cdot)$ is a frequency modulation characteristic and $F(\cdot)$ is an amplitude modulation characteristic. It is assumed that both modulation functions are bounded from above and strictly greater than zero.

The model in (6.6)-(6.8) is referred to as the pulse-modulated Smith model and was analyzed in detail in [30]. It constitutes a hybrid system that evolves in continuous time according to (6.6), but undergoes instantaneous jumps in the continuous state vector at discrete instants governed by (6.7). Each jump in the state vector corresponds to the secretion of a GnRH impulse.

By construction, this model will, just as the biological system, always have sustained oscillations. It thus solves the problem with stationary solutions in the classical Smith model discussed in Section 6.3.1. The

dynamics of (6.6)-(6.8) are thoroughly studied and are known to exhibit oscillating solutions that are either periodic or chaotic [169].

However, the model in (6.6)-(6.8) does not take into account the time it takes for LH, which is secreted in the pituitary, to travel through the bloodstream to the testes, where Te is produced. Therefore the pulse-modulated Smith model is extended with time delays in Chapter 9, where we also adopt techniques from system identification in order to estimate the unknown parameters from clinical data.

Chapter 7

Pulse-modulated feedback

The model of testosterone regulation in Section 6.3.3 implements pulse-modulated feedback control. In this chapter, we study this type of feedback mechanism in some more detail and give a review of the linear finite-dimensional case. We then go on investigating how infinite-dimensional dynamics, e.g. time-delays and averaging over a sliding window, can be incorporated into the model. It is shown that when the infinite-dimensional dynamics are described by certain pseudodifferential operators, most of the analysis performed for the finite-dimensional case carries over directly to the infinite-dimensional case.

7.1 Introduction

A typical feedback control system is schematically depicted in Figure 7.1. It consists of a dynamical system $\mathcal{S}$, whose output is desired to follow a certain reference signal. This can be achieved by a feedback controller $\mathcal{C}$ that uses the reference signal and the output signal in order to compute a suitable input signal.

The field of control theory started to emerge in the first half of the 20th century [1]. Today, feedback control is an integral part of most engineered systems. A reason for the success of feedback control is that it can give control strategies that perform well even though the components of the system are not in perfect shape, e.g. due to wear and tear. This property is of course also desirable when it comes to the functions inside the human body, so it should come as no surprise that evolution has equipped humans with numerous feedback mechanisms, from the cell level to the level of a complete organism. An example of such system making use of multiple feedback loops is the endocrine system.

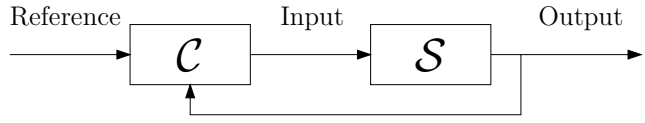


Figure 7.1: Block diagram of a feedback control system

In pulse-modulated feedback, the controller $\mathcal{C}$ feeds the system with a signal that consists of a train of pulses fired at discrete time instants. Due to, among other things, simple realization and low power consumption, this type of control has been used in fields such as electrical, space, heating and pneumatic engineering [52]. Due to the pulsatile nature of neural networks and hormone secretion [157], it has also found application in modeling of biomedical systems [30].

7.2 Pulse-modulated feedback control in finite-dimensional linear models

In pulse-modulated feedback, different types of pulses, e.g. rectangular pulses, can be used. However, here we consider instantaneous impulses, described by the Dirac delta function. A wide range of signal shapes can then be obtained as the impulse response of a dynamical filter.

In this section, we consider models on the following form

$$\dot{x}(t) = Ax(t), \qquad \text{if } t \neq t_k, \tag{7.1}$$

$$x(t_k^+) = x(t_k^-) + w_k B, \qquad \text{if } t = t_k, \tag{7.2}$$

$$y(t) = Cx(t). \tag{7.3}$$

Note that this include, for example, the pulse-modulated model for testosterone regulation presented in Section 6.3.3. The controller task is to decide the impulse times t_k and the impulse weights w_k. The time between impulses, sometimes called the clock interval, will be denoted by $T_k = t_{k+1} - t_k$.

There are several established control strategies in pulse-modulated feedback. On the one hand, in amplitude modulation, the clock intervals T_k are predetermined to have uniform length, while the amplitude w_k of each pulse is determined by the control mechanism. In frequency modulation, on the other hand, the length of each clock interval T_k is calculated by the controller, while the amplitudes are predetermined.

According to biological evidence, both the amplitude and frequency of the pulses are manipulated by the controller in testosterone regulation, so this is the approach that will be studied below.

In the literature, a distinction is often made between two types of pulse-modulated control. A first possibility is that the length of the

clock interval T_k and the corresponding amplitude w_k are determined directly by the output in the beginning of each interval. That is,

$$t_{k+1} = t_k + T_k, \quad T_k = \Phi(y(t_k^-)), \quad w_k = F(y(t_k^-)), \tag{7.4}$$

for some static functions $\Phi(\cdot)$ and $F(\cdot)$. This is refereed to as *modulation of the first kind*. As can be seen from (6.8), this is the strategy used in the model of Section 6.3.3.

However, more elaborate types of modulation are possible. For example, the next clock interval $T_k = t_{k+1} - t_k$ could be determined as the minimal positive root of the equation

$$T_k = \Phi(\sigma(T_k + t_k)),$$

where $\sigma(t)$ is some function that depends on the output in the interval $[0, t]$. In this way, the controller can get more information than just the output value at one time instant in order to make a decision. This is called *modulation of the second kind*. An example of this is that the impulse could occur when the output goes below a certain threshold.

Below, we consider modulation of the first kind. One reason for this is that it simplifies the analysis of the dynamical model. Also, when an autonomous closed-loop system is considered, the behavior of the system between impulse times can in principle be encoded into the modulation functions $\Phi(\cdot)$ and $F(\cdot)$. Hence, concentrating on modulation of the first kind is not a severe restriction for our purposes.

7.2.1 Periodic solutions

Assuming that $\Phi(\cdot)$ and $F(\cdot)$ are strictly positive and bounded, the model described by (7.1)-(7.4) is a self-sustained oscillating model, since there will always be a new pulse within a finite time. Thus it is of interest to study oscillatory, e.g. periodic, solutions. In [52], stability and oscillations in pulse-modulated models are investigated. Periodical solutions to the specific model given by (7.1)-(7.4) are analyzed in [30], by means of the mapping

$$Q(x) = e^{A\Phi(Cx)}(x + F(Cx)B). \tag{7.5}$$

Let $x(t)$ be a solution to (7.1)-(7.4) and $x_k = x(t_k^-)$ for all $k \geq 0$. Then, given x_0, we can compute x_k for all $k > 0$ by iterating the mapping

$$x_k = Q(x_{k-1}), \quad k \geq 1.$$

Furthermore, if we are given the sequence $x_0, x_1, \ldots$, then we can use (7.1) to reconstruct $x(t)$ for all times $t \geq t_0$.

Definition 7.1. *Let m be some integer $m \geq 1$. A solution $x(t)$ to* (7.1)-(7.4) *is called an m-cycle if $x(t)$ is periodic and has exactly m impulses in the least period.*

Denote the mth iteration of the mapping as $Q^{(m)}(x)$, i.e.,

$$Q^{(1)}(x) = Q(x), \quad Q^{(m)}(x) = Q(Q^{(m-1)}(x)), \quad m > 1.$$

Assume that the equality

$$x_0 = Q^{(m)}(x_0),$$

holds, and that m is the smallest integer for which it holds, then it can be seen that the solution $x(t)$ with initial value $x(t_0^-) = x_0$ is an m-cycle. Hence, periodical solutions $x(t)$ can be studied in terms of the discrete map $Q(x)$.

Definition 7.2. *Consider a solution $x(t)$ to* (7.1)-(7.4)*, and define $x_k = x(t_k^-)$. Let $\tilde{x}(t)$ be a perturbed solution with the initial value $\tilde{x}_0 = \tilde{x}(t_0^-)$ and $\tilde{x}_k = \tilde{x}(t_k^-)$. Then the solution $x(t)$ is called* orbitally stable *if, for any $\varepsilon > 0$, there exists $\varepsilon_0 > 0$ such that if $\|x_0 - \tilde{x}_0\|_2 < \varepsilon_0$ then $\|\tilde{x}_n - x_n\|_2 < \varepsilon$ for all $n \geq 0$.*

The solution $x(t)$ will be called asymptotically orbitally stable *if it is orbitally stable, and moreover, there exists a number $\varepsilon_1 > 0$ such that $\|\tilde{x}_n - x_n\|_2 \to 0$ when $n \to \infty$ if $\|\tilde{x}_0 - x_0\|_2 < \varepsilon_1$.*

Stability in the sense of Definition 7.2 can be established by examining the Jacobian matrix $J(x)$ of $Q(x)$. Note that, due to the chain rule, the Jacobian of the mth iteration $Q^{(m)}(x)$ is given by

$$J^{(m)}(x_0) = J(x_{m-1}) \cdots J(x_0),$$

where $x_k = Q(x_{k-1})$ as above.

Theorem 7.1. *Consider an m-cycle $x(t)$ and let $x_k = x(t_k^-)$. Then the periodic solution is asymptotically orbitally stable if $J^{(m)}(x_0)$ is Schur stable, i.e. all its eigenvalues lie inside the unit circle in the complex plane.*

Proof. This is a well known result, see e.g. [32] and [4]. □

7.3 Pseudodifferential operators

In this section, a class of pseudodifferential operators is introduced. Such operators will be used in Section 7.5 in order to augment the pulse-modulated model in (7.1)-(7.3) with infinite-dimensional dynamics that

can represent time delays, averaging etc. These operators can also be useful in estimation of the impulses, as we will see in Chapter 8.

Let $X(s)$ be the Laplace-transform of some signal $x(t)$. Then the operator P defined by

$$(Px)(t) = \mathcal{L}^{-1}\{p(s)X(s)\} = \frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty} p(s)X(s)e^{st}ds, \tag{7.6}$$

where c is a suitable real constant, is called a pseudodifferential operator with symbol $p(s)$ [42]. Note that the differential operator d/dt is a special case of pseudodifferential operators, and has the symbol $p(s) = s$.

7.3.1 Finite-memory operators

We will call a causal operator P with symbol $p(s)$ a finite-memory (FM) operator with memory length τ, if $(Px)(t)$ is invariant to $x(r)$ for $r < t - \tau$. For example, the operator $(Px)(t) = x(t)$ with symbol $p(s) = 1$ has memory length 0, and the time-delay operator $(Px)(t) = x(t - \tau)$ with symbol $p(s) = e^{-s\tau}$ has memory length τ.

We are here interested in operators with finite, but non-zero, memory length. In particular, we assume that the symbol $p(s)$ satisfies the following assumption.

Assumption 7.1. *The symbol $p(s)$ is an entire function, and there exists $\tau, K \geq 0$ such that,*

$$|p(s)| \leq Ke^{\tau|\operatorname{Re}\{s\}|}, \tag{7.7}$$

for all $s \in \mathbb{C}$. Furthermore assume that

$$\lim_{s\to\infty} p(s) = 0. \tag{7.8}$$

Note that it follows by the Paley-Wiener theorem that an operator that satisfies (7.7) has compact support, see e.g. [146]. Hence, any such operator is an FM-operator, and the memory length is given by the smallest τ for which the inequality in the assumption holds.

Furthermore, if $p(s)$ satisfies (7.7) with $\tau = 0$, then it is an entire and bounded function and thus constant. However, the only constant function that satisfies (7.8) is $p(s) = 0$. We can thus conclude that, if an operator P with symbol $p(s) \neq 0$ satisfies Assumption 7.1, then it is an FM-operator with a memory length strictly larger than zero.

Lemma 7.1. *Assume that the operators P_1 and P_2 with the corresponding symbols $p_1(s)$ and $p_2(s)$, and memory lengths τ_1 and τ_2, both satisfy Assumption 7.1. Then the following statements hold.*

Table 7.1: Examples of operators with memory length τ, that satisfy Assumption 7.1, and their corresponding matrix functions. Here $f(t)$ is a scalar piecewise continuous function and A is a non-singular matrix.

$(\boldsymbol{P}x)(t)$	$p(s)$	$p(A)$
$x(t-\tau)$	$e^{-s\tau}$	$e^{-A\tau}$
$\int_0^\tau x(t-r)dr$	$\frac{1-e^{-s\tau}}{s}$	$\left(I-e^{-A\tau}\right)A^{-1}$
$\int_0^\tau f(r)x(t-r)dr$	$\int_0^\tau f(t)e^{-st}dt$	$\int_0^\tau f(t)e^{-At}dt$

- *The operator P with symbol $p(s) = p_1(s)p_2(s)$ satisfies Assumption 7.1 and has a memory length $\tau \leq \tau_1 + \tau_2$.*
- *The operator P with symbol $p(s) = p_1(s) + p_2(s)$ satisfies Assumption 7.1 and has memory length $\tau \leq \max\{\tau_1, \tau_2\}$.*

Proof. If $|p_1(s)| \leq K_1 e^{\tau_1 |\operatorname{Re}\{s\}|}$ and $|p_2(s)| \leq K_2 e^{\tau_2 |\operatorname{Re}\{s\}|}$ for all $s \in \mathbb{C}$, and they are entire then both $p_1(s)p_2(s)$ and $p_1(s) + p_2(s)$ are entire. Furthermore

$$|p_1(s)p_2(s)| \leq K e^{\tau |\operatorname{Re}\{s\}|},$$

with $K = K_1 K_2$ and $\tau = \tau_1 + \tau_2$, showing the first case of the lemma.

For the second case, note that

$$\begin{aligned} |p_1(s) + p_2(s)| &\leq |p_1(s)| + |p_2(s)| \leq K_1 e^{\tau_1 |\operatorname{Re}\{s\}|} + K_2 e^{\tau_2 |\operatorname{Re}\{s\}|} \\ &\leq K e^{\tau |\operatorname{Re}\{s\}|}, \end{aligned}$$

with $K = 2\max\{K_1, K_2\}$ and $\tau = \max\{\tau_1, \tau_2\}$.

Finally, if $p_1(s)$ and $p_2(s)$ satisfy (7.8) then

$$\begin{aligned} \lim_{s\to\infty} (p_1(s)p_2(s)) &= \lim_{s\to\infty} p_1(s) \lim_{s\to\infty} p_2(s) = 0, \\ \lim_{s\to\infty} (p_1(s) + p_2(s)) &= \lim_{s\to\infty} p_1(s) + \lim_{s\to\infty} p_2(s) = 0. \end{aligned}$$

□

It is straightforward to show that if $p(s)$ is the finite Laplace transform [36] of some piecewise continuous function $f(t)$, i.e.

$$p(s) = \int_0^\tau f(t)e^{-st}\, dt,$$

then $p(s)$ satisfies Assumption 7.1, and has memory length (at most) τ. Some popular operators that satisfy Assumption 7.1 are shown in Table 7.1, together with their corresponding matrix function $p(A)$, which will play a crucial role in the derivations of our main result.

7.3.2 Functions of matrices

Consider a scalar function $p : \mathbb{C} \to \mathbb{C}$, and a square matrix A. What would we then mean by $p(A)$? There are several ways to define the matrix $p(A)$: e.g. through power series, diagonalization of the matrix, Jordan decomposition or a Cauchy integral. While the mentioned approaches produce functions $p(A)$ that are defined on different domains (e.g., diagonalization only works for diagonalizable matrices), they all yield the same matrix when $p(A)$ is defined [68]. Since we here are interested in functions $p(s)$ that are entire, we will define $p(A)$ through the Cauchy integral.

Let Γ be any simple closed rectifiable curve that strictly encloses all of the eigenvalues of $A \in \mathbb{C}^{n\times n}$. For a function $p(s)$ that is analytic on Γ, we define the matrix function $p(A)$ as

$$p(A) = \frac{1}{2\pi i} \oint_\Gamma p(s)(sI - A)^{-1}\, ds. \tag{7.9}$$

Given $p(s)$, it might be cumbersome to evaluate $p(A)$ using (7.9), but the following theorem can often be used in order to simplify the computations.

Theorem 7.2. *Consider a square matrix A and assume that $p(s), g(s)$ and $h(s)$ are entire functions. Then the following relationships hold*

- *If $p(s) = 1$, then $p(A) = I$.*
- *If $p(s) = s^n$, then $p(A) = A^n$, where $n \in \mathbb{N}$.*
- *If $p(s) = e^{st}$, then $p(A) = e^{At}$, where $t \in \mathbb{R}$.*
- *If $p(s) = g(s) + h(s)$, then $p(A) = g(A) + h(A)$.*
- *If $p(s) = g(s)h(s)$, then $p(A) = g(A)h(A) = h(A)g(A)$.*
- *If $p(s) = g(s)/h(s)$ and $h(A)$ is nonsingular, then $p(A) = g(A)h(A)^{-1} = h(A)^{-1}g(A)$.*
- *$p(A)$ commutes with any matrix that commutes with A.*
- *For a nonsingular matrix T, $p(TAT^{-1}) = Tp(A)T^{-1}$.*
- *If $A = diag(\lambda_1, \ldots, \lambda_n)$, then $p(A) = diag(p(\lambda_1), \ldots, p(\lambda_n))$.*

Proof. See e.g. [68]. □

Remark 7.1. *Note that if $p(s) = g(s)/h(s)$ and $p(s)$ is an entire function then all the roots of $h(s)$ are roots of $g(s)$. This means that even if $h(A)$ is singular at certain points, the matrix function $p(A)$ is still well-defined. For example, if*

$$p(s) = \frac{1 - e^{(\lambda - s)\tau}}{s - \lambda},$$

then

$$p(A) = \begin{cases} \left(I - e^{(\lambda I - A)\tau}\right)(A - \lambda I)^{-1} & \text{if } \lambda \notin \sigma(A) \\ \sum_{k=1}^{\infty} \frac{\tau^k}{k!}(\lambda I - A)^{k-1} & \text{if } \lambda \in \sigma(A) \end{cases}.$$

The matrix exponential e^{At}, which is the matrix function corresponding to the exponential function e^{st}, is of special interest when studying linear state-space models. It is well known that the solution to the state-space model

$$\dot{x}(t) = Ax(t) \tag{7.10}$$

can be written as $x(t) = \Phi_A(t)x(0)$, where the transition matrix $\Phi_A(t)$ is given by the matrix exponential in the time-invariant case [128], i.e.

$$\Phi_A(t) = e^{At}.$$

The exponential function is usually defined by the power series,

$$e^{At} = \sum_{k=0}^{\infty} \frac{1}{k!}(tA)^k,$$

but, as discussed above, we can equivalently define it through the Cauchy integral

$$e^{At} = \frac{1}{2\pi i} \oint_{\Gamma} e^{st}(sI - A)^{-1} ds,$$

where Γ encloses all eigenvalues of A. If an FM-operator is applied to the solution $x(t)$, the following lemma is useful.

Lemma 7.2. *Let P be a pseudodifferential operator with the symbol $p(s)$ that satisfies Assumption 7.1 with the memory length τ. Then, for any square matrix A,*

$$(P\Phi_A)(t) = \mathcal{L}^{-1}\left\{p(s)(sI - A)^{-1}\right\}(t) = p(A)e^{At}, \tag{7.11}$$

for all $t > \tau$, where $\Phi_A(t) = e^{At}$.

Proof. See Appendix 7.A.1. □

It directly follows from Lemma 7.2 that, if the operator P satisfies Assumption 7.1 and $x(t)$ is the solution to (7.10), i.e. $x(t) = e^{At}x(0)$, then for $t > \tau$,

$$(Px)(t) = p(A)e^{At}x(0).$$

As an example, let P be the time-delay operator with the symbol $p(s) = e^{-s\tau}$. Then, for $t > \tau$, it follows that

$$(Px)(t) = p(A)e^{At}x(0) = e^{-A\tau}e^{At}x(0) = e^{A(t-\tau)}x(0) = x(t - \tau).$$

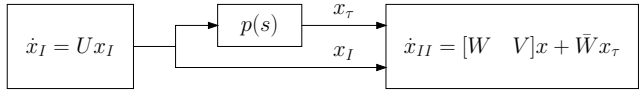


Figure 7.2: An FD-reducible model structure. Here $x = [x_I^\top \quad x_{II}^\top]^\top$, and $p(s)$ is a pseudodifferential operator satisfying Assumption 7.1.

7.4 Finite-dimensional reducibility

The goal of this chapter is to extend the pulse-modulated model in (7.1)-(7.4) with infinite-dimensional dynamics. We do this by first studying the autonomous model

$$\dot{x}(t) = A_0 x(t) + A_1 x_\tau(t), \tag{7.12}$$

$$x_\tau(t) = \begin{cases} \varphi(t) & \text{if } t \leq \tau \\ (Px)(t) & \text{if } t > \tau \end{cases}, \tag{7.13}$$

where P is a pseudodifferntial operator with the symbol $p(s)$ that satisfies Assumption 7.1. The initial function $\varphi(t)$ could be chosen equal to $(Px)(t)$, but the formulation in (7.13) gives the possibility for more general initial functions.

We want to consider cases where the infinite-dimensional dynamics can be reduced to finite-dimensional such, i.e. we want our model to be finite-dimensional (FD) reducible.

Definition 7.3. *The model described by* (7.12)-(7.13) *is called FD-reducible if there exists* $A \in \mathbb{R}^{n_x \times n_x}$ *such that any solution* $x(t)$ *of* (7.12)-(7.13) *satisfies*

$$\frac{dx}{dt} = Ax \tag{7.14}$$

for $t > \tau$.

When the operator P is given by $(Px)(t) = kx(t)$ for some constant k, then the model is finite-dimensional already, so we trivially get $\dot{x}(t) = (A_0 + kA_1)x$ for all $t \geq 0$ in (7.12). However, for operators with memory length $\tau > 0$, we have to impose restrictions on the matrices A_0 and A_1 in order to ensure that the model is FD-reducible.

Assumption 7.2. *The matrices* A_0 *and* A_1 *are both non-zero and satisfy*

$$A_1 A_0^k A_1 = 0,$$

for $k = 0, \ldots, n_x - 1$.

While Assumption 7.2 might seem a bit strong, it turns out that any model possessing a cascade structure such as in Figure 7.2 is FD-reducible. This is shown in the following lemma.

Lemma 7.3. *Assumption 7.2 is equivalent to the following statements:*
(i) There exists a nonsingular $n_x \times n_x$ matrix T such that

$$T^{-1}A_0T = \begin{bmatrix} U & 0 \\ W & V \end{bmatrix}, \quad T^{-1}A_1T = \begin{bmatrix} 0 & 0 \\ \overline{W} & 0 \end{bmatrix}, \tag{7.15}$$

where the blocks U and V are square matrices, and the blocks W and $\overline{W}$ are of the same dimension.
(ii) For any entire function $\tilde{p}(s)$, we have

$$A_1\tilde{p}(A_0)A_1 = 0.$$

Proof. The equivalence between Assumption 7.2 and (i) is shown in [33]. That Assumption 7.2 implies (ii) follows by expansion of $\tilde{p}(s)$ in a power series and the Cayley-Hamilton theorem. The fact that (ii) implies Assumption 7.2 follows by considering $\tilde{p}(s) = s^k$ for $k = 1, \ldots, n_x - 1$. □

Lemma 7.4. *If (7.12)-(7.13) satisfy Assumption 7.1 and Assumption 7.2, then the model is FD-reducible and the matrix A in (7.14) is given by*

$$A = A_0 + A_1p(A_0). \tag{7.16}$$

Proof. To see this, first notice that, for $t \geq 0$, $x(t)$ is given by [54]

$$x(t) = e^{A_0t}x(0) + \int_0^t e^{A_0(t-r)}A_1x_\tau(r)dr.$$

It thus follows from Lemma 7.3 that

$$A_1x(t) = A_1e^{A_0t}x(0), \quad A_1p(A_0)x(t) = A_1p(A_0)e^{A_0t}x(0).$$

Recalling that the operator P is a linear operator, we obtain

$$A_1(Px)(t) = A_1(P\Phi_{A_0})(t)x(0),$$

where $\Phi_{A_0}(t) = e^{A_0t}$. Hence it follows from Lemma 7.2 that, for $t > \tau$,

$$A_1x_\tau(t) = A_1p(A_0)e^{A_0t}x(0) = A_1p(A_0)x(t).$$

Inserting this into (7.12) gives $\dot{x}(t) = (A_0 + A_1p(A_0))x(t)$. □

The lemma demonstrates that the infinite-dimensional dynamics in (7.12) obeys a finite-dimensional model with the system matrix $A = A_0 + A_1p(A_0)$ for $t > \tau$. An important property of this matrix is highllighted in the following corollary to Lemma 7.3.

Corollary 7.1. *Assume that A_0 and A_1 satisfy Assumption 7.2. Let $A = A_0 + A_1 p(A_0)$, and $\tilde{p}(s)$ be any entire function. Then*

$$A_1\tilde{p}(A_0) = A_1\tilde{p}(A).$$

Proof. To see this, it is enough to show that $A_1A_0^k = A_1A^k$ for all $k \geq 0$. This clearly holds for $k = 0$. Assume that it holds that $A_1A_0^k = A_1A^k$, then

$$A_1A^{k+1} = A_1A_0^kA = A_1A_0^{k+1} + A_1A_0^kA_1p(A_0) = A_1A_0^{k+1},$$

where the last equality follows from case (ii) in Lemma 7.3. Hence, the corollary follows by induction. □

7.5 The extended pulse-modulated model

We are now ready to extend the pulse-modulated model in (7.1)-(7.4) with infinite-dimensional dynamics by considering the following system of equations:

$$\dot{x}(t) = A_0x(t) + A_1x_\tau(t), \qquad \text{if } t \neq t_k, \tag{7.17}$$

$$x(t_k^+) = x(t_k^-) + w_kB_0, \qquad \text{if } t = t_k, \tag{7.18}$$

$$y(t) = Cx(t), \tag{7.19}$$

$$x_\tau(t) = \begin{cases} \varphi(t) & \text{if } t \leq \tau \\ (Px)(t) & \text{if } t > \tau \end{cases}, \tag{7.20}$$

where P is an operator with the symbol $p(s)$ that satisfies Assumption 7.1 and has the memory length τ. Furthermore, we assume that the continuous part in (7.17) satisfies Assumption 7.2. The impulse weights and impulse times are, as before, given by

$$t_{k+1} = t_k + T_k, \quad T_k = \Phi(y(t_k)), \quad w_k = F(y(t_k)). \tag{7.21}$$

Without loss of generality we assume that the first impulse times is at $t_0 = 0$. Also assume that the modulation functions are bounded from below and above,

$$\tau < \Phi_1 \leq \Phi(\cdot) \leq \Phi_2 < \infty, \quad 0 < F_1 \leq F(\cdot) \leq F_2 < \infty.$$

Note that this assumptions implies $T_k > \tau$ for all $k \geq 0$, and hence there can only be one impulse at a time inside the memory of the operator P.

Lemma 7.5. *Assume that the model in* (7.17)-(7.21) *satisfies Assumption 7.1 and Assumption 7.2. Then, for $t_k < t < t_{k+1}$, $k \geq 1$, it follows that*

$$A_1x_\tau(t) = A_1\left(p(A_0)e^{A_0(t-t_k)}x(t_k^-) + w_k(P\Phi_{A_0})(t-t_k)B_0\right), \tag{7.22}$$

where $\Phi_{A_0}(t) = e^{A_0 t}$. *Furthermore, if the initial function* $\varphi(t)$ *satisfies for* $0 < t < \tau$

$$A_1\varphi(t) = A_1\left(p(A_0)e^{A_0 t}x(0^-) + w_0(P\Phi_{A_0})(t)B_0\right), \tag{7.23}$$

then (7.22) *also holds for* $k = 0$.

Proof. As in the proof of Lemma 7.4, we can see that for $t_{k-1} < t < t_{k+1}$, we have

$$A_1x(t) = A_1\left(e^{A_0(t-t_{k-1})}x(t_{k-1}^+) + w_k e^{A_0(t-t_k)}B_0H(t-t_k)\right),$$

where $H(t)$ is the Heaviside step function, so

$$A_1p(A_0)x(t_k^-) = A_1p(A_0)e^{A_0(t_k-t_{k-1})}x(t_{k-1}^+).$$

Hence, if $t_k < t < t_{k+1}$, then $t > t_{k-1} + \tau$, so it follows by Lemma 7.2 that

$$\begin{aligned} A_1(Px)(t) &= A_1\left(p(A_0)e^{A_0(t-t_{k-1})}x(t_{k-1}^+) + w_k(P\Phi_{A_0})(t-t_k)B_0\right) \\ &= A_1\left(p(A_0)e^{A_0(t-t_k)}x(t_k^-) + w_k(P\Phi_{A_0})(t-t_k)B_0\right). \end{aligned}$$

Since $A_1x_\tau(t) = A_1(Px)(t)$ for $t > \tau$, it follows that (7.22) holds for $k \geq 1$. If $\varphi(t)$ is chosen in accordance with (7.23), it is also clear that (7.22) will hold for $k = 0$, too. □

Note that choosing the initial function as in (7.23) can intuitively be seen as assuming that the system behaved according to (7.17) before $t = 0$.

Lemma 7.6. *Assume that the model in* (7.17)-(7.21) *satisfies Assumption 7.1 and Assumption 7.2, and that* $p(A_0)$ *is invertible. For* $k \geq 1$ *and* $t_k < t < t_{k+1}$, *the solution* $x(t)$ *is then given by*

$$x(t) = e^{A(t-t_k)}(x(t_k^-) + w_kB) + w_k\eta(t-t_k)p(A_0)^{-1}B_0, \tag{7.24}$$

where A *is defined by* (7.16), $B = p(A)p(A_0)^{-1}B_0$, *and*

$$\eta(t) = (P\Phi_A)(t) - (P\Phi_{A_0})(t) - \left(e^{At}p(A) - e^{A_0t}p(A_0)\right).$$

Furthermore, with the initial function chosen according to (7.23), *the above also holds for* $k = 0$.

Proof. See Appendix 7.A.2. □

7.5.1 Reduction to a finite-dimensional impulsive system

Since $\eta(t)$ in Lemma 7.6 is zero for $t > \tau$, we can see that (7.17) behaves as a finite-dimensional system most of the time. This is formalized in the following corollary.

Corollary 7.2. *Consider the finite-dimensional system*

$$\dot{\hat{x}}(t) = A\hat{x}(t), \qquad \text{if } t \neq \hat{t}_k, \tag{7.25}$$

$$\hat{x}(\hat{t}_k^+) = \hat{x}(\hat{t}_k^-) + \hat{w}_k B, \qquad \text{if } t = \hat{t}_k, \tag{7.26}$$

$$\hat{y}(t) = C\hat{x}(t), \tag{7.27}$$

$$\hat{t}_{k+1} = \hat{t}_k + \Phi(\hat{y}(\hat{t}_k^-)), \qquad \hat{w}_k = F(\hat{y}(\hat{t}_k^-)), \tag{7.28}$$

where $A = A_0 + A_1 p(A_0)$ *and* $B = p(A)p^{-1}(A_0)B_0$. *Let* $k \geq 1$, *and assume that* $\hat{t}_k = t_k$ *and* $\hat{x}(t_k^-) = x(t_k^-)$. *Then* $\hat{t}_n = t_n$, $\hat{w}_n = w_n$ *and* $\hat{x}(t_n^-) = x(t_n^-)$ *for all* $n \geq k$. *Moreover,*

$$\hat{x}(t) = x(t), \quad t_n + \tau < t < t_{n+1}.$$

Furthermore, if the initial function $\varphi(t)$ *satisfies* (7.23), *then the above is also true for* $k = 0$.

Proof. The solution to (7.25) is, for $\hat{t}_k < t < \hat{t}_{k+1}$, given by

$$\hat{x}(t) = e^{A(t-\hat{t}_k)}(\hat{x}(\hat{t}_k^-) + \hat{w}_k B).$$

From Lemma 7.2, we have that $\eta(t) = 0$ for $t > \tau$, and thus the corollary follows from Lemma 7.6. □

Using this corollary, most of the analysis performed for finite-dimensional system (7.1)-(7.3) carries over to the infinite-dimensional case when Assumption 7.1 and Assumption 7.2 are satisfied.

7.5.2 Periodic solutions

Let $x(t)$ be a solution of (7.17)-(7.21) and consider mapping (7.5), i.e.

$$Q(x) = e^{A\Phi(Cx)}(x + F(Cx)B),$$

with $A = A_0 + A_1 p(A_0)$ and $B = p(A)p(A_0)^{-1}B_0$. As before, define $x_k = x(t_k^-)$. It follows by Corollary 7.2 that

$$x_{k+1} = Q(x_k),$$

for $k \geq 1$. If the initial function satisfies (7.23), we also obtain $x_1 = Q(x_0)$. The opposite also holds, i.e. if Q generates the sequence $x_0, x_1, \ldots$, then there is a solution to (7.17)-(7.21) that fulfills $x(t_k^-) = x_k$ for all k.

Hence, the analysis performed for periodic solutions in the finite-dimensional case in Section 7.2.1 carries over to the study of periodic solutions to (7.17). In particular, an m-cycle in (7.17) is asymptotically stable if the Jacobian $J^{(m)}(x(t_0^-))$ is Schur stable.

7.6 Numerical examples

Now consider the model in Section 7.5 with the system matrices given by

$$A_0 = \begin{bmatrix} -b_1 & 0 & 0 \\ g_1 & -b_2 & 0 \\ 0 & 0 & -b_3 \end{bmatrix}, \qquad A_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & g_2 & 0 \end{bmatrix},$$
$$B_0^\top = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$

Note that $p(s) = 1$ implies $A = A_0 + A_1$ and the model becomes the one for testosterone regulation presented in Section 6.3.3. However, with the framework developed in this chapter, we can now introduce time-delays corresponding to the time it takes for LH to travel from the pituitary to the testes and other effects, see Chapter 9. For illustration purposes, we here use

$$x_\tau(t) = \int_0^\tau K(r)x(t-r)dr, \quad K(t) = e^{-\alpha t}/\beta,$$

where $\alpha \geq 0, \beta > 0$, which corresponds to a pseudodifferential operator P with the symbol

$$p(s) = \frac{1 - e^{-(s+\alpha)\tau}}{\beta(s+\alpha)}. \tag{7.29}$$

The model parameters are chosen as $b_1 = 0.4$, $b_2 = 0.01$, $b_3 = 0.045$, $g_1 = 2$ and $g_2 = 4$, and the modulation functions as

$$\Phi(y) = 50 + 220\frac{(y/r)^4}{1 + (y/r)^4}, \quad F(y) = 1.5 + \frac{5}{1 + (y/r)^4},$$

with $r = 100$. The parameters α and τ are varied, and the inverse gain β chosen as

$$\beta = \int_0^\tau e^{-\alpha s}ds = \begin{cases} \tau, & \alpha = 0 \\ (1 - e^{-\alpha\tau})/\alpha, & \alpha > 0 \end{cases}$$

for normalization. Note that with this normalization, we have

$$\lim_{\tau \to 0} p(s) = 1,$$

so when τ goes to zero, we arrive at the finite-dimensional case. The same is true when $\alpha \to \infty$ since

$$\lim_{\alpha \to \infty} p(s) = 1. \tag{7.30}$$

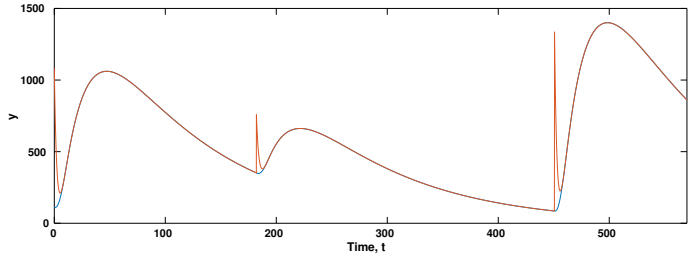


Figure 7.3: The output $y(t) = Cx(t)$ for the infinite-dimensional system (blue), and $\hat{y}(t) = C\hat{x}(t)$ for the corresponding finite-dimensional system (red).

7.6.1 Illustration of finite-dimensional reduction

In Corollary 7.2, it was shown that, given the infinite-dimensional model (7.17)-(7.21), there is a corresponding finite-dimensional model that behaves in the same way at all time instants except for a time interval of length τ immediately after each impulse. In the finite-dimensional model, the matrices are given by

$$A = A_0 + A_1 p(A_0), \quad B = p(A)p^{-1}(A_0)B_0.$$

For illustration, we will use $\alpha = 0.2$ and $\tau = 10$ in $p(s)$. Then

$$p(A_0) = \frac{1}{\beta}\left(I - e^{-(A_0+\alpha I)\tau}\right)(A_0 + \alpha I)^{-1} = \begin{bmatrix} 7.39 & 0 & 0 \\ -32.58 & 1.04 & 0 \\ 0 & 0 & 1.18 \end{bmatrix},$$

which gives

$$A = \begin{bmatrix} -0.4 & 0 & 0 \\ 2 & -0.01 & 0 \\ -130.44 & 4.14 & -0.046 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 277.36 \end{bmatrix}.$$

Since $CB_0 = 0$, the output of the infinite-dimensional model is continuous. However, for the corresponding finite-dimensional model, we have $CB \neq 0$, so the output of this model jumps at each impulse time.

The solution $y(t) = Cx(t)$ of (7.17) is plotted together with the output $\hat{y}(t) = C\hat{x}(t)$ of the corresponding finite-dimensional model in Figure 7.3. We can see that $\hat{y}(t)$ jumps after each impulse time, and yet it holds that $\hat{y}(t) = y(t)$ after 10 time units.

7.6.2 Bifurcations

In order to explore what type of solutions the system might possess, bifurcation diagrams were obtained for different values of α by varying the memory length τ.

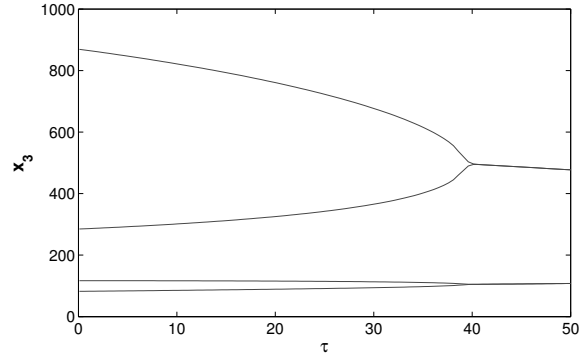


Figure 7.4: Bifurcation diagram for $\alpha = 0$.

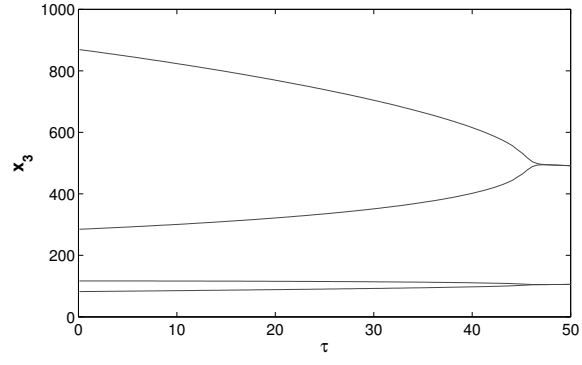


Figure 7.5: Bifurcation diagram for $\alpha = 0.02$.

Figure 7.4–Figure 7.6 depict the bifurcation diagrams obtained by varying τ from 0 to 50. It can be seen that for $\tau = 0$ (i.e. $p(s) = 1$) the system has a stable 4-cycle. With an increase in the value of τ, the 4-cycle is then reduced to a 2-cycle for $\alpha = 0$ (mean value over a sliding window of width τ) and $\alpha = 0.02$. This demonstrates that increasing the distributed delay leads here to a simplification of the nonlinear dynamics, i.e. less oscillating solutions. This is in a sharp contrast with the conventional view on the role of time delays as a destabilizing factor. A similar effect has also been observed for the pointwise time-delay operator in [29].

The figures present the simulation results only for $\tau \leq 50$, since $\tau < \Phi_1$ should be satisfied for the discrete mapping to be valid. However, it is of course possible to simulate the system for larger time-delays. Then it can be seen that the 2-cycle for large τ is, for $\alpha = 0$, further reduced to a 1-cycle. However, for the cases of $\alpha = 0.02$ and $\alpha = 0.2$, the effect of

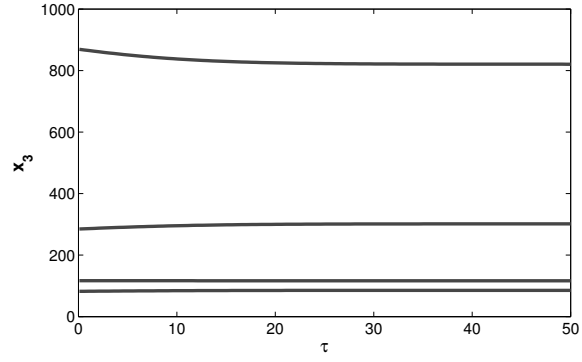


Figure 7.6: Bifurcation diagram for $\alpha = 0.2$.

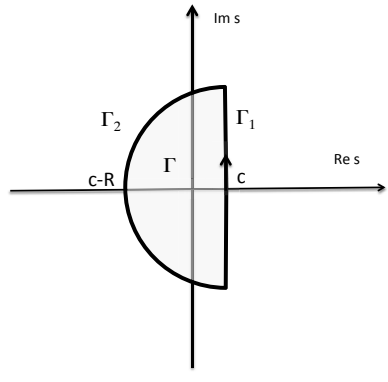


Figure 7.7: The contour Γ in the complex s-plane comprising the line Γ_1 and the semicircle Γ_2.

increasing the time delay is not noticeable in the same way. Figure 7.6 demonstrates that high decay rates of the kernel function render the system dynamics fairly insensitive to the time-delay phenomena, which is to be expected from (7.30).

7.A Proofs

7.A.1 Proof of Lemma 7.2

Let c be a constant greater than the real part of any eigenvalue of A, and consider the contour in Figure 7.7. Note that

$$\mathcal{L}^{-1}\left\{p(s)(sI-A)^{-1}\right\}(t) = \lim_{R\to\infty}\frac{1}{2\pi i}\int_{\Gamma_1} p(s)(sI-A)^{-1}e^{st}\,ds,$$

which also implies that

$$\mathcal{L}^{-1}\left\{p(s)(sI-A)^{-1}\right\}(t) = \lim_{R\to\infty}\frac{1}{2\pi i}\left(\oint_{\Gamma} p(s)(sI-A)^{-1}e^{st}\,ds - \int_{\Gamma_2} p(s)(sI-A)^{-1}e^{st}\,ds\right). \tag{7.31}$$

We first consider the line integral along Γ_2. Note that the integral is taken with respect to each element in the matrix, and the absolute value of the largest element is smaller than the infinity norm of the matrix. Using the change of variables $s = c + Re^{i\theta}$, for each element of the matrix, we have

$$\left|\int_{\Gamma_2} p(s)(sI-A)^{-1}_{i,j}e^{st}ds\right| \le \int_{\Gamma_2}|p(s)|\left\|(sI-A)^{-1}\right\|_\infty|e^{st}||ds| \le K\int_{\Gamma_2}\left\|(sI-A)^{-1}\right\|_\infty e^{\tau|\operatorname{Re}\{s\}|}|e^{st}||ds| =$$

$$KR\int_{\pi/2}^{3\pi/2}\|(cI+Re^{i\theta}I-A)^{-1}\|_\infty e^{\tau|c+R\cos\theta|}e^{t(c+R\cos\theta)}d\theta. \tag{7.32}$$

First note that

$$e^{\tau|c+R\cos\theta|}e^{t(c+R\cos\theta)} \le e^{(t+\tau)c}e^{(t-\tau)R\cos\theta}, \quad \frac{\pi}{2}\le\theta\le\frac{3\pi}{2}.$$

For the matrix norm, assume that $R > |c - a_{kk}|$ and notice that the diagonal elements of $sI - A$ are given by

$$|s-a_{kk}| = |c+Re^{i\theta}-a_{kk}| \ge R+K_2, \quad \frac{\pi}{2}\le\theta\le\frac{3\pi}{2}.$$

where $K_2 = a_{kk} - c$ if $c \ge a_{kk}$ and $K_2 = 0$ otherwise. Hence, for large enough R, the matrix $sI - A$ will be diagonally dominant and, in this case, as shown in [152], it applies that

$$\|(sI-A)^{-1}\|_\infty \le \frac{1}{\alpha},$$

where

$$\alpha = \min_k\left(|s-a_{kk}| - \sum_{j\neq k}a_{kj}\right).$$

We can thus conclude that for large enough R, it holds that

$$\|(sI-A)^{-1}\|_\infty \le \frac{1}{R+K_2},$$

where K_2 is a constant that is invariant to R and θ. Inserting these inequalities into (7.32) thus gives

$$\left|\int_{\Gamma_2} p(s)(sI-A)^{-1}_{i,j}e^{st}\,ds\right| \le \frac{KRe^{c(t+\tau)}}{R+K_2}\int_{\pi/2}^{3\pi/2} e^{(t-\tau)R\cos\theta}\,d\theta =$$

$$\frac{2KRe^{c(t+\tau)}}{R+K_2}\int_0^{\pi/2} e^{-(t-\tau)R\sin\theta}\,d\theta \le \frac{2KRe^{c(t+\tau)}}{R+K_2}\int_0^{\pi/2} e^{-2(t-\tau)R\theta/\pi}\,d\theta =$$

$$\frac{\pi Ke^{c(t+\tau)}}{R+K_2}(1-e^{-(t-\tau)R}) \to 0, \quad \text{as } R\to\infty,$$

where the last inequality follows from Jordan's inequality since $t>\tau$.

Inserting this into (7.31) gives, for $t>\tau$ and R large enough,

$$\mathcal{L}^{-1}\left\{p(s)(sI-A)^{-1}\right\}(t) = \frac{1}{2\pi i}\oint_\Gamma p(s)e^{st}(sI-A)^{-1}ds = p(A)e^{At},$$

where the last equality follows from (7.9) and Theorem 7.2.

7.A.2 Proof of Lemma 7.6

We prove this by taking the derivative of (7.24). First note that it follows from (7.8) in Assumption 7.1 and the initial value theorem that

$$(P\Phi_A)(0) = \lim_{s\to\infty} sp(s)(sI-A)^{-1} = 0. \tag{7.33}$$

Hence we see that

$$\lim_{t\to 0}\eta(t) = p(A_0)-p(A), \tag{7.34}$$

so it follows that

$$\begin{aligned}\mathcal{L}\left\{\frac{d}{dt}\eta(t)\right\} &= \eta(0) + sp(s)\left((sI-A)^{-1}-(sI-A_0)^{-1}\right)\\ &\quad - s\left(p(A)(sI-A)^{-1}-p(A_0)(sI-A_0)^{-1}\right)\\ &= p(s)\left(A(sI-A)^{-1}-A_0(sI-A_0)^{-1}\right)\\ &\quad -\left(Ap(A)(sI-A)^{-1}-A_0p(A_0)(sI-A_0)^{-1}\right),\end{aligned}$$

where the last equality follows by using $s(sI-M)^{-1} = M(sI-M)^{-1}+I$, which holds for any square matrix M. Hence, with $A = A_0 + A_1p(A_0)$, we get

$$\begin{aligned}\dot\eta(t) &= A\left(P\Phi_A\right)(t) - A_0\left(P\Phi_{A_0}\right)(t) - \left(Ae^{At}p(A) - A_0e^{A_0t}p(A_0)\right)\\ &= A_0\eta(t) + A_1p(A_0)\left((P\Phi_A)(t) - p(A)e^{At}\right).\end{aligned}$$

From this, and Corollary 7.1, it follows that the derivative of (7.24) is given by

$$\begin{aligned}\dot{x}(t) &= A_0x(t) + A_1\left(p(A_0)e^{A_0(t-t_k)}x(t_k^-) + w_k(P\Phi_{A_0})(t-t_k)B_0\right)\\ &= A_0x(t) + A_1x_\tau(t),\end{aligned}$$

where the last equality follows from Lemma 7.5, and thus also holds for $k = 0$ if the initial function is chosen in accordance with (7.23). Hence, (7.24) satisfies (7.17). Furthermore, letting $t \to t_k$ in (7.24) we see from (7.34) that

$$x(t_k^+) = x(t_k^-) + w_kB,$$

so (7.24) also satisfies (7.18), and we can conclude that $x(t)$ given by (7.24) is the unique solution to (7.17)-(7.18).

Chapter 8

Blind estimation in impulsive systems

8.1 Introduction

In Chapter 7, we discussed pulse-modulated control in linear systems. If the control signal, i.e. the timing and weights of the pulses, is known, then the state of the system can be estimated using an observer in the same way as for any other type of input signal. However, in e.g. biomedical models, the impulsive control signal is often unknown. A characteristic example of this is the pulsatile feedback control of endocrine systems [45]. For example, in the model of testosterone regulation presented in Section 6.3.3, the pulses corresponds to the release of GnRH in the hypothalamus. Since it is not possible to measure the concentration of GnRH in a non-invasive way [79], we typically have to consider the control signal to be unknown.

State estimation in a system with an unknown impulsive input signal is challenging since the state variables are reset after a finite time. Thus, most classical asymptotic observers, such as the Luenberger observer [94], lose track of the state vector after each impulse. A similar phenomenon occurs with respect to state estimation in switching systems, when the switching time is unknown, as discussed in e.g. [148]. Impulsive observers can be utilized to properly deal with the state reset problem.

An impulsive observer that finds the true state in finite time when the input is known is presented in [124], and a similar approach is used to implement observer-based control for systems with persistently acting impulsive input in [168]. In [110], observers for impulsive systems with linear continuous-time dynamics and a linear resetting law are considered. Furthermore, in [32] and [31], a static gain observer for linear

continuous systems under impulsive feedback is studied, and conditions for local stability of the observer under periodic solutions in the plant are proved.

Unfortunately, most of the existing approaches to impulsive observer design fail when there is no information about the input impulses.

In this chapter, we present the impulsive model considered in Section 8.2. Here, the control signal and resetting law are assumed to be unknown. As a solution to the problem of estimating the timing and weights of the pulses, a finite memory operator presented in Section 8.3 is utilized. In Section 8.4, we see how this operator can be used in order to estimate the time and weight of each impulse and, in Section 8.5, the operator is used together with an impulsive observer in order to estimate both the pulses and the states of the system.

8.2 The impulsive model

Consider the following impulsive SISO state-space model

$$\dot{x}(t) = Ax(t), \qquad \text{if } t \notin \mathcal{T} \tag{8.1}$$

$$x(t^+) = x(t^-) + w_k B \qquad \text{if } t \in \mathcal{T} \tag{8.2}$$

$$y(t) = Cx(t), \tag{8.3}$$

where $A \in \mathbb{R}^{n_x \times n_x}, B \in \mathbb{R}^{n_x \times 1}$, $C \in \mathbb{R}^{1 \times n_x}$ and $\mathcal{T} = \{t_1, t_2, \ldots\}$ is a countable subset of $[0, \infty)$. The instants t_k are called impulse times, and are assumed to be ordered so that $t_1 < t_2 < t_3 < \ldots$. To each impulse time t_k there is a corresponding impulse weight w_k.

Note that the model (8.1)-(8.2) can equivalently be rewritten as

$$\dot{x}(t) = Ax(t) + B\xi(t),$$

where

$$\xi(t) = \sum_{t_k \in \mathrm{T}} w_k \delta(t - t_k),$$

and $\delta(\cdot)$ is the Dirac delta function. This reformulation clarifies why t_k, w_k are called impulse time and weight, respectively.

It is assumed that there is a minimum dwell time Φ, i.e.

$$0 < \Phi \leq t_{k+1} - t_k$$

for all k. This is a standard assumption in pulse-modulated systems. In this chapter the impulse times and weights are considered to be unknown, and the presented methods could easily be extended to handle known inputs and more than one output.

8.3 The finite-memory convolution operator

In this chapter, the finite-memory convolution operator

$$(Pv)(\lambda,\tau;t)=\int_{t-\tau}^{t}e^{\lambda(t-r)}f(r)dr \tag{8.4}$$

will be utilized. This is an FM-operator with memory length τ, cf. Section 7.3. The symbol of the operator $(Pv)(\lambda,\tau;t)$ is

$$p(\lambda,\tau,s)=\frac{1-e^{(\lambda-s)\tau}}{s-\lambda}, \tag{8.5}$$

Given symbol (8.5), the corresponding matrix function is evaluated in a closed form as

$$p(\lambda,\tau,M)\triangleq\left(I-e^{(\lambda I-M)\tau}\right)(M-\lambda I)^{-1},$$

if $M-\lambda I$ is invertible. Note that, letting $v(t)=Ce^{Mt}x_0$, it follows from Theorem 7.2, that

$$\begin{aligned}(Pv)(\lambda,\tau;t)&=\int_{t-\tau}^{t}e^{\lambda(t-r)}Ce^{Mr}x_0dr\\&=Cp(\lambda,\tau,M)e^{Mt}x_0\end{aligned} \tag{8.6}$$

for $t>\tau$. Now let $\Lambda=\{\lambda_1,\ldots,\lambda_m\}$ be a set of real and distinct elements, and introduce the following notation

$$W(t,M)\triangleq\begin{bmatrix}Cp(\lambda_1,t,M)\\\vdots\\Cp(\lambda_m,t,M)\end{bmatrix}.$$

Also let

$$\mathcal{V}(\tau,M)\triangleq W^{\top}(\tau,M)W(\tau,M),\quad\text{and}\quad V(t,M)\triangleq W(t,M)e^{Mt}. \tag{8.7}$$

Lemma 8.1. *If M has distinct eigenvalues, the pair (M,C) is observable, and $\sigma(M)\cap\Lambda=\emptyset$, then $\mathcal{V}(\tau,M)\succ 0$.*

Proof. See [113]. □

8.3.1 Continuous least squares observers

In [111], a wide class of continuous least-squares observers based on FM-operators was introduced and studied. Thanks to the deadbeat properties of these observers, they have also been proved useful in, e.g.

fault detection [113]. In this section, we see how such an observer can be constructed from the operator in (8.4), and in Section 8.4 we apply it in order to estimate the impulses.

For notational convince we let

$$\mathcal{V}_\tau \triangleq \mathcal{V}(\tau, A)$$

and

$$Y(\lambda, \tau; t) \triangleq (Py)(\lambda, \tau; t).$$

Assumption 8.1. *The matrix A has distinct eigenvalues and (A, C) is an observable pair. Furthermore, Λ is chosen so that it contains at least n_x distinct elements and $\sigma(A) \cap \Lambda = \emptyset$.*

If Assumption 8.1 is satisfied, then it follows from Lemma 8.1 that $\mathcal{V}_\tau$ is invertible, so we can introduce the observer

$$\hat{x}_\tau(t) = \mathcal{V}_\tau^{-1} \sum_{\lambda_i \in \Lambda} p^\top(\lambda_i, \tau, A) C^\top Y(\lambda_i, \tau; t). \tag{8.8}$$

Lemma 8.2. *If Assumption 8.1 holds and $\mathcal{T} \cap [t-\tau, t] = \emptyset$, then $\hat{x}_\tau(t) - x(t) = 0$.*

Proof. See [111]. □

Remark 8.1. *Other types of deadbeat observers can be constructed in a similar way as in* (8.8)*, by replacing the operator $(Py)(\lambda, \tau; t)$ with other FM-operators, see e.g. [114].*

Lemma 8.2 states that, if there is no impulse within the memory (sliding window) of (8.4), then the observer estimate $\hat{x}_\tau(t)$ is equal to the true state $x(t)$. The following lemma describes what happens to the estimate when an impulse enters the sliding window.

Lemma 8.3. *If Assumption 8.1 holds and $\mathcal{T} \cap [t-\tau, t] = \{t_k\}$, then*

$$\hat{x}_\tau(t) = e^{A\tau} x(t-\tau) + w_k \mathcal{V}_\tau^{-1} W^\top(\tau, A) W(t - t_k, A) e^{A(t-t_k)} B,$$

while the true state vector is given by

$$x(t) = e^{A\tau} x(t-\tau) + w_k e^{A(t-t_k)} B.$$

Proof. See Appendix 8.A.1 □

8.3.2 Properties of the symbol

Lemma 8.2 proves that observer (8.8) has deadbeat performance for almost every set Λ, but it does not give any suggestion for how to choose Λ. In [114], the sensitivity of (8.8) to uncertainty in the system matrix of the plant is studied using the Fréchet derivative. In [10] and [112], the effect of measurement noise on the observer estimate is investigated, providing at least some insight into the choice of Λ.

In this section, the influence of the user parameters on the low-pass filter characteristics of $p(\lambda, \tau, s)$ is discussed. Consider

$$\bar{p}(\lambda, \tau, s) = \frac{\lambda}{e^{\lambda\tau} - 1} p(\lambda, \tau, s),$$

i.e. the operator $p(\lambda, \tau, s)$ with static gain normalized to one. Define ω_b as the bandwidth of $\bar{p}(\lambda, \tau, s)$, i.e. the lowest frequency such that the inequality

$$|\bar{p}(\lambda, \tau, j\omega)| < \frac{1}{\sqrt{2}}, \quad \forall \omega > \omega_b$$

is satisfied. Then

$$|\bar{p}(\lambda, \tau, j\omega)|^2 = \frac{1 + e^{2\lambda\tau} - 2e^{\lambda\tau}\cos(\omega\tau)}{\omega^2 + \lambda^2} \frac{\lambda^2}{(e^{\lambda\tau} - 1)^2} \leq \frac{1 + e^{2\lambda\tau} + 2e^{\lambda\tau}}{\omega^2 + \lambda^2} \frac{\lambda^2}{(e^{\lambda\tau} - 1)^2} = \frac{\lambda^2}{\omega^2 + \lambda^2} \coth^2(\lambda\tau/2)$$

and thus an upper bound for the bandwidth is

$$\omega_b \leq \bar{\omega}_b \triangleq \sqrt{\lambda^2(2\coth^2(\lambda\tau/2) - 1)}.$$

For a fixed τ, it can be shown that

$$\inf_{\lambda} \bar{\omega}_b = \lim_{\lambda \to 0} \sqrt{\lambda^2(2\coth^2(\lambda\tau/2) - 1)} = \frac{2\sqrt{2}}{\tau}.$$

The above result suggests that observer (8.8) is better at filtering out high frequency noise when the elements of Λ are small and τ is large. However, in general, choosing all $\lambda_i \in \Lambda$ close to zero may lead to numerical problems. The choice of τ is also a trade-off, since the estimation delay increases with τ. Also, for the purposes of impulse estimation, τ has to be less than the minimal dwell time Φ, see Section 8.4.

8.4 Pulse estimation

In order to detect when an impulse occurs, two observers of the form (8.8) with overlapping sliding windows can be used. A similar approach, involving two deadbeat observers, was used for fault detection in [113].

In this section, we consider the case when Assumption 8.1 holds. Choose the time delays τ_1 and τ_2 so that

$$0 < \tau_1 < \tau_2 < \Phi.$$

This particular choice of τ_1 and τ_2 ensures that there can be at most one impulse within the time interval $[t - \tau_2, t]$.

Lemma 8.4. *If Assumption 8.1 holds, then given the two time delays τ_1 and τ_2, there is a state space realization of* (8.1)-(8.3) *such that*

$$\mathcal{V}_{\tau_1} = I, \quad \mathcal{V}_{\tau_2} = F \tag{8.9}$$

where F is a diagonal matrix with strictly positive elements.

Proof. See Appendix 8.A.2. □

In what follows, it is assumed that the state vector in the plant equations (8.1) is chosen so that (8.9) holds. This not only provides more compact notation but also optimizes the numerical properties of the underlying least-squares observers, [111].

Now introduce the two observers $\hat{x}_{\tau_1}(t)$ and $\hat{x}_{\tau_2}(t)$, as in (8.8). The difference between the estimates produced by the observers is called the state estimation residual and is denoted as

$$r(t) = \hat{x}_{\tau_1}(t) - \hat{x}_{\tau_2}(t).$$

It follows that $r(t) = 0$ when there is no impulse within $[t - \tau_2, t]$, and that $r(t) \neq 0$ otherwise. Next, two ways to extract information about t_k and w_k from $r(t)$ are considered.

8.4.1 Integrating the state estimation residual

Note that if there is an impulse at time t_k, then thanks to the dwell time, there is no impulse in $[t_k - \Phi, t_k)$ or $(t_k, t_k + \Phi]$. Hence, for any $0 < \epsilon < \Phi$, we have

$$R(t_k) \triangleq \int_{t_k}^{t_k+\tau_2} r(\theta)\, d\theta = \int_{t_k-\epsilon}^{t_k+\tau_2+\epsilon} r(\theta)\, d\theta. \tag{8.10}$$

We can thus compute the integral $R(t_k)$ without knowing exactly when the resetting event occurs, and in the following lemma an analytic expression for $R(t_k)$ is given.

Lemma 8.5. *If $t_k \in \mathcal{T}$, then*

$$R(t_k)/w_k = \left(e^{A\tau_2} - e^{A\tau_1}\right) A^{-1}B + \mathcal{V}_{\tau_1}^{-1} W^\top(\tau_1, A)K(\tau_1, A)B - \mathcal{V}_{\tau_2}^{-1} W^\top(\tau_2, A)K(\tau_2, A)B,$$

where

$$K(\tau, A) = \begin{bmatrix} K_1(\tau, A) \\ \vdots \\ K_m(\tau, A) \end{bmatrix},$$

$$K_i(\tau, A) = C\left((e^{A\tau} - I)A^{-1} - \frac{1}{\lambda_i}(e^{\lambda_i \tau} - 1)I\right)(A - \lambda_i I)^{-1}.$$

Proof. See Appendix 8.A.3 □

Note that $R(t_k)/w_k$ is a constant that does not depend on the impulse time t_k and that we can compute this constant given the system matrices and the observers.

Next we define the vectors a_k by

$$a_0 = 0, \quad \text{and } a_k = R(t_k) + a_{k-1} \text{ for } k > 0.$$

Since $r(t) = 0$ for $t \in (t_k + \tau_2, t_{k+1})$, it follows that

$$a_k = \int_0^t r(s)\, ds,$$

for all $k > 0$ and $t \in [t_k + \tau_2, t_{k+1}]$. When the output signal is corrupted by measurement noise, the constants a_k can be estimated by taking the mean value over an interval where the integral is nearly constant, see Example 8.1. Since

$$a_k - a_{k-1} = R(t_k)$$

and $R(t_k)/w_k$ is a known constant, the estimates of $a_k - a_{k-1}$ can be used to estimate w_k.

8.4.2 Estimating impulse times

Assume that the values of $Y(\lambda_j, \tau_2; t)$ are computed for at least two $\lambda_j \in \Lambda$ involved in the estimate $\hat{x}_{\tau_2}(t)$. Then the result of the following proposition can be used in order to estimate both t_k and w_k.

Lemma 8.6. *If $\mathcal{T} \cap [t - \tau_2, t] = \{t_k\}$ and $\mathcal{T} \cap [t - \tau_1, t] = \emptyset$ then*

$$w_k e^{(t-t_k)\lambda} C(\lambda I - A)^{-1} B = Y(\lambda, \tau_2; t) - C(\lambda I - A)^{-1}(e^{\lambda\tau_2}\hat{x}_{\tau_1}(t - \tau_2) - \hat{x}_{\tau_1}(t)). \quad (8.11)$$

Proof. Along the lines of Appendix 8.A.1, it can be seen that

$$Y(\lambda, \tau_2; t) = C(\lambda I - A)^{-1} \left(w_k e^{\lambda(t-t_k)} B + e^{\lambda \tau_2} x(t-\tau_2) - x(t) \right).$$

This, together with the fact that $\hat{x}_{\tau_1}(t-\tau_2) = x(t-\tau_2)$ and $\hat{x}_{\tau_1}(t) = x(t)$ when $\mathcal{T} \cap [t-\tau_2, t] = \{t_k\}$ and $\mathcal{T} \cap [t-\tau_1, t] = \emptyset$, completes the proof. □

Note that the system in the two unknowns t_k and w_k

$$\begin{cases} w_k e^{(t-t_k)\lambda_1} = f_1 \\ w_k e^{(t-t_k)\lambda_2} = f_2 \end{cases}$$

is uniquely solved by

$$\begin{cases} t_k = \frac{\ln(f_2) - \ln(f_1)}{\lambda_1 - \lambda_2} + t \\ \\ w_k = f_1 e^{(t_k - t)\lambda_1} \end{cases}.$$

Since each row in (8.11) can be written in a similar manner, w_k and t_k can be evaluated exactly as long as $Y(\lambda_j, \tau_2; t)$ is computed for two distinct λ_j, which is the case considered below. However, it is also possible to compute $Y(\lambda_j, \tau_2; t)$ for all $\lambda_j \in \Lambda$, and solve the resulting overdetermined system of equations, e.g. using least-squares techniques. This can be implemented in the form of the algorithm below.

1. Start while $r(t)$ is zero.
2. Wait for $r(t)$ to become non-zero, implying that a resetting event has occurred.
3. Wait for time τ_1. Now the expression in Proposition 8.6 holds.
4. Compute the right-hand side of (8.11) for at least two λ and solve for t_k and w_k. It is also possible to use estimates of w_k computed as in Section 8.4.1 and solve only for t_k.

Note that it is not necessary to obtain the exact time when $r(t)$ becomes non-zero. In fact, it suffices that Step 3 of the algorithm is performed at a time $t \in (t_k + \tau_1, t_k + \tau_2)$. Thus, in order to reduce sensitivity to noise, one can carry out Step 3 for several instants t and then use the mean value as an estimate. In practice, the check for a non-zero state estimate residual in Step 1 can be replaced by a check against a suitable threshold.

Example 8.1: Estimation of input impulses

Here the techniques proposed in Section 8.4.1-8.4.2 are tested on a numerical example.

Assume the following values in model (8.1)-(8.3),

$$A = \begin{bmatrix} -0.1 & 0 \\ 0.8 & -0.014 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix}, \qquad x(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top,$$

with four impacting impulses at the time instants $\mathcal{T} = \{5, 80, 160, 230\}$ and with the weights $w_1 = 0.8$, $w_2 = 0.9$, $w_3 = 0.6$ and $w_4 = 0.7$. This is the same structure as in the GnRH-LH part of the pulsatile Smith model for testosterone regulation in Section 6.3.3.

To evaluate the performance of the estimation algorithms under measurement noise, zero mean white noise with variance 0.1 is added to the output of (8.1)-(8.3). The output of the plant with noise is shown in Figure 8.1. For the observer, the time delays are assigned as $\tau_1 = 30$

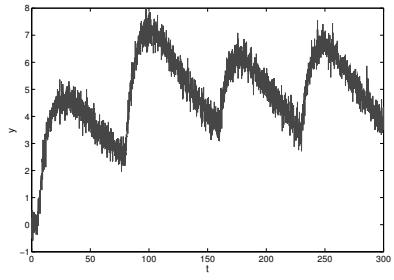


Figure 8.1: The output signal of Example 8.1 with measurement noise

and $\tau_2 = 50$, while $\Lambda = \{-0.03, -0.08\}$. Balancing the state-space description with the transformation $x \to Tx$ with

$$T = \begin{bmatrix} -758.49 & 27.13 \\ 163.24 & -1.45 \end{bmatrix},$$

makes the system satisfy (8.9) with

$$F = \begin{bmatrix} 0.18 & 0 \\ 0 & 597.39 \end{bmatrix}.$$

Choosing the elements of Λ closer to zero would reduce the impact of the noise, but would also lead to a larger condition number of F. For the balanced system, Lemma 8.5 shows that

$$\frac{R(t_k)}{w_k} = \begin{bmatrix} -354.85 \\ 192.24 \end{bmatrix}. \tag{8.12}$$

From the discussion in Section 8.4.1, it follows that the integral of $r(t)$ in the noise-free case is constant on the intervals $(t_k + \tau_2, t_{k+1})$. The integral of $r(t)$ corrupted by noise can be seen in Figure 8.2. The constants a_k are approximated by taking the mean of the red parts in Figure 8.2, resulting in the estimated values

$$\hat{a}_1 = \begin{bmatrix} -282.99 \\ 152.99 \end{bmatrix}, \quad \hat{a}_2 = \begin{bmatrix} -602.17 \\ 326.25 \end{bmatrix}$$
$$\hat{a}_3 = \begin{bmatrix} -814.32 \\ 441.11 \end{bmatrix}, \quad \hat{a}_4 = \begin{bmatrix} -1062.3 \\ 575.52 \end{bmatrix}.$$

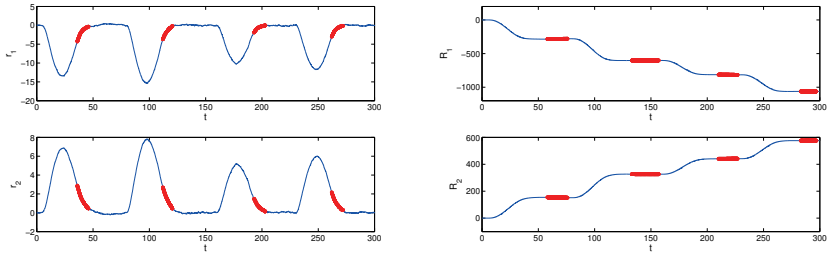


Figure 8.2: *Left:* The state estimation residual $r(t)$. *Right:* The integral of $r(t)$. The parts marked with a red line are used to compute w_k and t_k.

Let W_k be the result of element-wise division of $\hat{a}_k - \hat{a}_{k-1}$ by the right-hand side of (8.12). Taking the mean of the elements in W_k as estimates for w_k gives

$$\hat{w}_1 = 0.7955, \quad \hat{w}_2 = 0.9016, \quad \hat{w}_3 = 0.5977, \quad \hat{w}_4 = 0.6990.$$

Next, both the resetting times t_k and the impulse weights w_k are estimated using the procedure in Section 8.4.2. The function $r(t)$ is plotted in Figure 8.2. In an attempt to decrease the sensitivity to noise, the right hand side of (8.11) has been computed for all t marked by a red line in Figure 8.2. This gives an estimate of w_k and t_k for each t, and taking the mean value of all these estimates yields

$$\begin{aligned} \hat{t}_1 &= 5.22, & \hat{t}_2 &= 80.06, & \hat{t}_3 &= 159.87, & \hat{t}_4 &= 230.12, \\ \hat{w}_1 &= 0.8101, & \hat{w}_2 &= 0.9104, & \hat{w}_3 &= 0.5937, & \hat{w}_4 &= 0.6968. \end{aligned}$$

The estimates of w_k using this method are slightly worse than those for the integration method, but here estimates of the resetting times t_k are also obtained.

8.5 Blind state estimation

In this section, an algorithm for state estimation in (8.1)-(8.3) is derived. Since the input impulses are unknown, the state estimation problem is called blind. The proposed method is based on a linear impulsive observer and the FM convolution operator in (8.4).

8.5.1 The observer

In order to estimate the state vector of (8.1), the impulsive observer

$$\frac{d\hat{x}}{dt} = A\hat{x} + K(y - \hat{y})\,, \qquad t \notin \hat{\mathcal{T}} \tag{8.13}$$

$$\hat{x}(t^+) = \hat{x}(t^-) + \hat{w}_l B\,, \qquad t = \hat{t}_l \in \hat{\mathcal{T}} \tag{8.14}$$

$$\hat{y}(t) = C\hat{x}(t) \tag{8.15}$$

is used, where $\hat{t}_l, \hat{w}_l$ are referred to as the observer impulse times and weights respectively. Let

$$D = A - KC.$$

Note that, if the pair (A, C) is observable, then so is (D, C). In this section, we will apply the operator (8.4) to the observer output $\hat{y}(t)$ instead of directly to the true output $y(t)$, so in order to make use of Lemma 8.1 we need the following assumption.

Assumption 8.2. *(A, C) is an observable pair and the matrix $D = A - KC$ has distinct eigenvalues. Furthermore, Λ is chosen so that it contains at least n_x distinct elements and $\sigma(D) \cap \Lambda = \emptyset$.*

The main difference between Assumption 8.1 and Assumption 8.2 is that the second case is concerned with D instead of A. Since the eigenvalues of D can be freely chosen by the user, this is a less restrictive assumption.

The state estimation error $\varepsilon(t) = x(t) - \hat{x}(t)$ is governed by the equations

$$\frac{d\varepsilon}{dt} = D\varepsilon\,, \qquad t \notin \mathcal{T} \cup \hat{\mathcal{T}}, \tag{8.16}$$

$$\varepsilon(t^+) = \varepsilon(t^-) + w_k B\,, \qquad t = t_k \in \mathcal{T}, \tag{8.17}$$

$$\varepsilon(t^+) = \varepsilon(t^-) - \hat{w}_l B\,, \qquad t = \hat{t}_l \in \hat{\mathcal{T}}. \tag{8.18}$$

If the impulses in the plant are known, then the observer impulses could be chosen so that $\hat{t}_k = t_k$ and $\hat{w}_k = w_k$. In this case, the state estimation error for $t \geq 0$ is given by $\varepsilon(t) = e^{Dt}\varepsilon(0)$.

However, in the case investigated in here, the plant impulses are unknown. To solve the problem with unknown impulses, it is assumed that, at any time t, the future output values $y(\cdot)$ within a sliding window $y(\theta), \theta \in [t, t + \tau)$ are made available to the observer. This is possible e.g. when the observer is run off-line or the state estimates are allowed to be delayed τ time units. A periodic mode in the plant, as in e.g. [32], also opens up for an application of the present technique.

Furthermore, it is assumed that $\tau < \Phi$, so that there is at most one plant impulse in the sliding window at any time t.

At time t, the observer applies operator (8.4) to the output $y(\theta)$, $\theta \in [t, t+\tau)$, to decide whether or not an observer impulse should be added in the current interval. If so, the same operator is used to evaluate the observer impulse time and weight.

8.5.2 Impulsive observer algorithm

We first give a general outline of the algorithm, and then discuss the details of each step.

1. Propagate the observer according to (8.13), until the condition for adding an observer impulse is met. Let t_o be the time when the condition was met.
2. Determine the observer impulse time $\hat{t}_l \in [t_o, t_o+\tau)$ and weight $\hat{w}_l$.
3. Propagate the observer according to (8.13)-(8.14) until $t = t_o + \tau$.
4. Go to Step 1.

In order to choose the observer impulses, operator (8.4) is applied to the output error $\bar{y}(t)$ that would be present if there were no observer impulses after some time t_o. That is

$$\bar{y}(t) = C\bar{\varepsilon}(t),$$

where $\bar{\varepsilon}(t)$ is the solution to (8.16)-(8.17) with the initial value $\bar{\varepsilon}(t_o) = \varepsilon(t_o)$. Define

$$Q(t_o) \triangleq \begin{bmatrix} (P\bar{y})(\lambda_1, \tau; t_o+\tau) \\ \vdots \\ (P\bar{y})(\lambda_m, \tau; t_o+\tau) \end{bmatrix}. \tag{8.19}$$

Notice that $\bar{y}(t)$, and thus $Q(t)$, can be computed if $\hat{x}(t)$ and the output $y(r)$ are known for $r \in [t, t+\tau)$.

Lemma 8.7. *If* $\mathcal{T} \cap [t, t+\tau) = \{t_k\}$, *then*

$$Q(t) = V(\tau, D)\varepsilon(t) + w_k V(t+\tau-t_k, D)B,$$

where $V(t, D)$ *is defined in* (8.7). *If* $\mathcal{T} \cap [t, t+\tau) = \emptyset$, *then*

$$Q(t) = V(\tau, D)\varepsilon(t).$$

Proof. Follows along the lines of Appendix 8.A.1. □

Lemma 8.7 shows in what way $Q(t)$ is affected when a plant impulse enters the sliding window $[t, t+\tau)$.

Lemma 8.8. *Assume that* $\mathcal{T} \cap [t_o, t_o + \tau) = \{t_k\}$ *and* $\hat{\mathcal{T}} \cap [t_o, t_o + \tau) = \{\hat{t}_l\}$. *Then*

$$\begin{aligned} W(\tau, D)\varepsilon(t_o + \tau) &= V(\tau, D)\varepsilon(t_o) \\ &+ w_k \left(V(t_o + \tau - t_k, D)B + E(t_k - t_o)\right) \\ &- \hat{w}_l \left(V(t_o + \tau - \hat{t}_l, D)B + E(\hat{t}_l - t_o)\right), \end{aligned} \tag{8.20}$$

where

$$E(\hat{t}) = (W(\tau, D) - W(\tau - \hat{t}, D))e^{D(\tau - \hat{t})}B.$$

Furthermore, for any $\epsilon, r > 0$, *it is possible to choose the set* Λ *such that each row of* $V(t_o + \tau - \hat{t}, D)B$ *and* $E(\hat{t} - t_o)$ *satisfy*

$$|E_i(\hat{t} - t_o)| \leq r|V_i(t_o + \tau - \hat{t}, D)B|$$

when $\hat{t} - t_o \leq \tau - \epsilon$.

Proof. See Appendix 8.A.4 □

Lemma 8.8 justifies the following important approximation. If Λ is chosen such that the lemma holds for a small enough r and there are no impulses in $[t_o + \tau - \epsilon, t_o + \tau)$, then the terms $w_k E(t_k - t_o)$ and $\hat{w}_l E(\hat{t}_l - t_o)$ in (8.20) are negligible. Thus it follows from Lemma 8.7 that

$$W(\tau, D)\varepsilon(t_o + \tau) \approx Q(t_o) - \hat{w}_l V(t_o + \tau - \hat{t}_l, D). \tag{8.21}$$

Next we discuss the Step 1 - Step 2 in the proposed method.

Step 1: Deciding on observer impulse
In this step, the observer propagates the state estimate $\hat{x}(t)$ according to (8.13) and computes $Q(t)$, defined in (8.19), for each t. This continues until some time t_o, when it is decided that there should be an observer impulse in the interval $[t_o, t_o + \tau)$.

In order to see whether or not an observer impulse should be added, $\|Q(t)\|_2$ is considered. Note that if $\varepsilon(t) = 0$ and $\mathcal{T} \cap [t, t + \tau) = \emptyset$, then it follows from Lemma 8.7 that $Q(t) = 0$. However, as soon as an impulse at t_k enters the sliding window $[t, t+\tau)$, the norm of $Q(t)$ starts to increase according to the relationship

$$\|Q(t)\|_2 = |w_k| \, \|V(t + \tau - t_k, D)B\|_2, \quad 0 < t_k - t < \tau. \tag{8.22}$$

Thus, if $\|Q(t)\|_2 > 0$, then $\varepsilon(t) \neq 0$ and/or $\mathcal{T} \cap [t, t + \tau) \neq \emptyset$. In both cases, there is a reason to add an observer impulse. If $\varepsilon(t_o) \neq 0$, the observer impulse could be used to reduce the state estimation error and, if there is a plant impulse within $[t, t+\tau)$, then the observer should counter it with an observer impulse.

Hence, a condition for choosing t_o is that $\|Q(t_o)\| > 0$. For robustness sake, zero in the right-hand side of the inequality can be replaced by a threshold $\eta > 0$, i.e.

$$\|Q(t_o)\|_2 > \eta.$$

It is also desirable, as will be seen in the discussion of Step 2, that (8.21) holds at time t_o. Due to Lemma 8.8, it is thus preferable to choose t_o so that there is no plant impulse in $[t_o + \tau - \epsilon, t_o + \tau)$ and the designer should take this into account when picking a value for η.

When a bound on $|w_k|$ is available, as in the case of the pulse-modulated model studied in Chapter 7, this information can be used in order to choose η. Let w_M be such that $|w_k| \leq w_M$ for all k, then a good choice of η is

$$\eta > w_M \left\| V(\tau - \hat{t}, D)B \right\|, \quad \text{for } \tau - \epsilon < \hat{t} < \tau,$$

cf. (8.22). However, if η is chosen too large, small impulses might be missed.

It is often possible to devise a more advanced condition for selecting t_o by studying the function plot of

$$\|V(\tau - t_k, D)B\|, \quad 0 < t_k < \tau.$$

In this manner, it might be possible to choose t_o so that it does not depend on the impulse weights. Example 8.2 illustrates this approach.

Step 2: Evaluating the observer impulse

Suppose that it has been decided at time t_o that there should be an observer impulse in $[t_o, t_o + \tau)$. Let $\varepsilon_o = \varepsilon(t_o)$. Assume also that there is a plant impulse with the weight w_k at time $t_k \in [t_o, t_o + \tau)$. If it is not the case, then $w_k = 0$ throughout this section.

When $\varepsilon_o = 0$, it follows from Proposition 8.7 that t_k and w_k can be found by solving

$$Q(t_o) = \hat{w}_l V(t_o + \tau - \hat{t}_l, D)B. \tag{8.23}$$

However, for $\varepsilon_o \neq 0$, the above equation may not have a solution. Therefore, the following optimization problem is solved instead

$$\begin{aligned} \min_{\hat{t}_l, \hat{w}_l} \quad & \left\| Q(t_o) - \hat{w}_l V(t_o + \tau - \hat{t}_l, D)B \right\|_2, \\ \text{s.t} \quad & t_o < \hat{t}_l < t_o + \tau. \end{aligned} \tag{8.24}$$

Note that the solution for $\hat{w}_l$ is given by

$$\hat{w}_l = \left(V(t_o + \tau - \hat{t}_l, D)B \right)^{\dagger} Q(t_o).$$

Inserting this into (8.24), we get a non-convex optimization problem in $\hat{t}_l$. Since $\hat{t}_l$ is constrained to lie inside the interval $(t_o, t_o + \tau)$, a possible solution is to grid over this interval, and choose the solution that yields the least cost function value.

Note that, when $\varepsilon_o = 0$, the optimum of (8.24) is attained when $\hat{t}_l = t_k$ and $\hat{w}_l = w_k$. To analyze what happens when $\varepsilon_o \neq 0$, assume that an observer impulse is fired at $\hat{t}_l$. If Λ has been chosen so that Lemma 8.8 holds for a small enough r, and there is no impulse in $[t_o + \tau - \epsilon, t_o + \tau)$, then it can be concluded from (8.21) that

$$\|\varepsilon(t_o + \tau)\|^2_{\mathcal{V}(\tau,D)} \approx \left\|Q(t_o) - \hat{w}_l V(t_o + \tau - \hat{t}_l, D)B\right\|_2^2. \tag{8.25}$$

The right-hand side of (8.25) is exactly the quantity that is minimized in (8.24). Note that if $\hat{t}_l = t_k$ and $\hat{w}_l = w_k$, then it follows from Lemma 8.7 that

$$Q(t_o) - \hat{w}_l V(t_o + \tau - \hat{t}_l, D)B = V(\tau, D)\varepsilon(t) = V(\tau, D)e^{D\tau}\varepsilon_o.$$

So, if we use the solution to (8.24), then

$$\left\|Q(t_o) - \hat{w}_l V(t_o + \tau - \hat{t}_l, D)B\right\|_2 \leq \left\|e^{D\tau}\varepsilon_o\right\|_{\mathcal{V}(\tau,D)}.$$

Hence, if $\varepsilon_o \neq 0$, the observer might add an impulse that does not correspond to an impulse in the plant. However, in this case the "false" impulse is chosen so that

$$\|\varepsilon(t_o + \tau)\|_{\mathcal{V}(\tau,D)} \lesssim \left\|e^{D\tau}\varepsilon_o\right\|_{\mathcal{V}(\tau,D)},$$

where the right-hand side is the state estimation error that is acquired when the observer impulse coincides with the plant impulse.

In practice, the observer usually adds "false" impulses when the state estimation error is large, resulting in a faster convergence. As the state estimation error decays over time, the observer impulses will get closer to the true plant impulses, as seen in Example 8.2.

8.5.3 The observer parameters

The designer has to choose the observer gain matrix K, the length of the sliding window τ, and the set Λ.

The proof of Lemma 8.8 shows that letting the elements of Λ tend to negative infinity ensures that the approximation in (8.21) is valid. However, other aspects have also to be taken into account. In Section 8.3.2, it is seen that the operator in general is less sensitive to measurement noise when $|\lambda|$ is small. Thus there is a trade-off between minimizing r in Lemma 8.8 and decreasing the sensitivity to noise. Fortunately, this

trade-off can usually be handled in practice, as shown in Example 8.2. It is also seen that the sensitivity to high frequency noise is reduced when τ is increased. However, increasing τ leads to a longer lag in the state estimate. Furthermore, τ must be chosen less than Φ, which is set by the plant characteristics.

Finally, the observer gain K has to be selected. This will mainly affect the state estimates behavior in between impulses, and standard techniques for linear observers can be used here.

Example 8.2: Blind state estimation

The following values are chosen in (8.1)-(8.3) for this example

$$A = \begin{bmatrix} -0.08 & 0 & 0 \\ 2 & -0.15 & 0 \\ 0 & 0.5 & -0.2 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$
$$C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \qquad x_o = \begin{bmatrix} 4 & 20 & 60 \end{bmatrix}^\top,$$

with four impacting impulses

$$t_1 = 10, \quad t_2 = 40, \quad t_3 = 80, \quad t_4 = 140,$$
$$w_1 = 4, \quad w_2 = 5, \quad w_3 = 7, \quad w_4 = 6.$$

The observer is applied to sampled data with fast sampling (0.01 time units between each sample), to imitate continuous execution. To evaluate the effect of measurement noise, white noise with zero mean and variance 10 was added to the output. The system output $y(t)$ is shown in Figure 8.3.

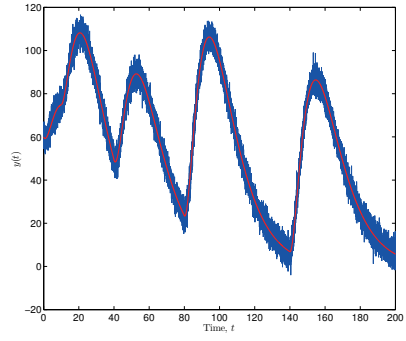


Figure 8.3: The output signal of the plant in Example 8.2 corrupted by noise (blue line) and without noise (red line).

For the observer, the gain was chosen as

$$K = \begin{bmatrix} 0.0002 \\ 0.0048 \\ 0.0200 \end{bmatrix},$$

so that the eigenvalues of D are placed at $-0.1, -0.15$ and -0.2. The length of the sliding window was set to $\tau = 20$. Finally, Λ was chosen as

$$\Lambda = \{-2.65, -2.70, -2.75, -2.80, -2.85, -2.90\}. \tag{8.26}$$

For this set of parameters, it holds that

$$|E_i(\hat{t} - t_o)| < 10^{-7}|V_i(t_o + \tau - \hat{t}, D)B|,$$

when $\hat{t} - t_o \leq 15$, cf. Lemma 8.8. Thus, if the optimization problem in (8.24) is only solved when $\mathcal{T} \cap [t_o + 15, t_o + 20) = \emptyset$, then (8.25) is a quite good approximation.

Next a condition for adding an observer impulse should be chosen. In light of the discussion in Section 8.5.2, first consider $\|V(t_o + \tau - t_k, D)B\|$, which is plotted in Figure 8.4. The maximum in Figure 8.4 occurs when $t_k - t_o \approx 5.8$, i.e. well below 15. This suggests choosing the optimization

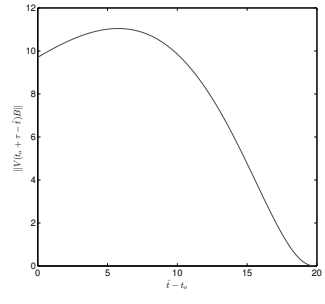


Figure 8.4: $\|V(t_o + \tau - t_k, D)B\|$ for $0 < t_k - t_o < \tau$.

instants t_o in the observer so that the quantity $\|Q(t)\|_2$ is at, or near, a maximum. Also, with this condition, the chosen optimization instants do not depend on the impulse weights. Hence, in this example, Step 1 of the algorithm in Section 8.5.2 will continue until $\|Q(t)\|_2$ reaches a maximum.

First the case of the noise-free output was tested. In Figure 8.5, the state estimates produced by the observer initialized with

$$\hat{x}(0) = \begin{bmatrix} 7 \\ 30 \\ 70 \end{bmatrix}$$

are shown together with the true values of the plant states. Also, the state estimation error is compared with the quantity $e^{Dt}(x(0) - \hat{x}(0))$, i.e. the state estimation error that we would get from an observer with all observer impulses being identical to the plant impulses. It can be seen that the proposed observer is completely off on the first impulse,

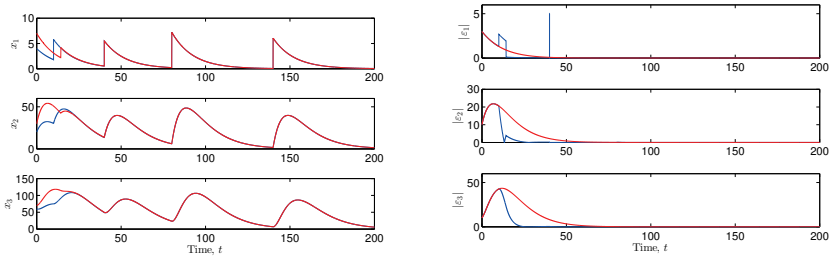


Figure 8.5: *Left:* True states $x(t)$ (blue line), and estimated states $\hat{x}(t)$ (red line). *Right:* State estimation error (blue line), and the state estimation error for an observer with exact knowledge of the plant impulses (red line).

but yet converges to the true state vector faster than it would if the first impulse were identical to the true plant impulse.

Figure 8.6 shows the result when measurement noise is added to the output. It can be seen that the estimated impulses are quite far from the true ones because of the measurement noise. In order to reduce the sensitivity to noise, we follow the discussion in Section 8.5.3 and move the elements of Λ closer to the origin, and choose instead

$$\Lambda = \{-0.65, -0.70, -0.75, -0.80, -0.85, -0.90\}. \tag{8.27}$$

The result is also shown in Figure 8.6, and this choice of Λ clearly get us closer to the true plant impulses.

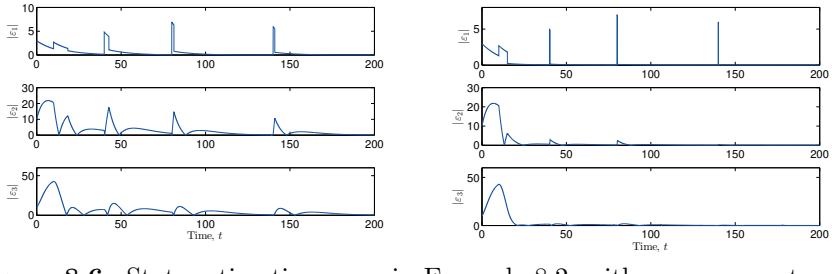


Figure 8.6: State estimation error in Example 8.2, with measurement noise. *Left:* With Λ given by (8.26). *Right:* With Λ given by (8.27).

8.A Proofs

8.A.1 Proof of Lemma 8.3

We first derive an analytic expression for $Y(\lambda, \tau; t)$ when $\mathcal{T} \cap [t-\tau, t] = \{t_k\}$. Let $x_o = x(t-\tau)$, then the state vector is given by

$$x(r) = \begin{cases} e^{A(r-t+\tau)}x_o & \text{if } r \in [t-\tau, t_k), \\ e^{A(r-t+\tau)}x_o + w_k e^{A(r-t_k)}B & \text{if } r \in (t_k, t]. \end{cases}$$

Hence, as in (8.6), it can be seen that

$$\begin{aligned} Y(\lambda, \tau; t) &= \int_{t-\tau}^{t} e^{\lambda(t-r)} C x(r)\, dr \\ &= \int_{t-\tau}^{t} e^{\lambda(t-r)} C e^{A(r-t+\tau)} x_o\, dr + w_k \int_{t_k}^{t} e^{\lambda(t-r)} C e^{A(r-t_k)} B\, dr \\ &= Cp(\lambda, \tau, A) e^{A\tau} x_o + w_k Cp(\lambda, t-t_k, A) e^{A(t-t_k)} B. \end{aligned}$$

It follows that

$$\begin{bmatrix} Y(\lambda_1, \tau; t) \\ \vdots \\ Y(\lambda_m, \tau; t) \end{bmatrix} = W(\tau, A) e^{A\tau} x_o + w_k W(t-t_k, A) e^{A(t-t_k)} B,$$

and thus

$$\hat{x}_\tau(t) = e^{A\tau} x(t-\tau) + w_k \mathcal{V}_\tau^{-1} W^\top(\tau, A) W(t-t_k, A) e^{A(t-t_k)} B.$$

8.A.2 Proof of Lemma 8.4

Let $\mathcal{V}_{\tau_1}$ and $\mathcal{V}_{\tau_2}$ correspond to the original system. A state transformation $\bar{x} = Tx$ results in

$$\overline{\mathcal{V}}_{\tau_1} = T^{-\top} \mathcal{V}_{\tau_1} T^{-1}, \quad \overline{\mathcal{V}}_{\tau_2} = T^{-\top} \mathcal{V}_{\tau_2} T^{-1}.$$

A transformation T such that $\overline{\mathcal{V}}_{\tau_1} = I$ and $\overline{\mathcal{V}}_{\tau_2} = F$ can be found in the following way. Since $\mathcal{V}_{\tau_1}$ is symmetric and positive definite there is an orthogonal U and diagonal L such that

$$\mathcal{V}_{\tau_1}^{-1} = ULU^\top.$$

Let $T_1 = L^{-1/2} U^\top$ and $\overline{\mathcal{V}} = T_1 \mathcal{V}_{\tau_2}^{-1} T_1^\top$. $\overline{\mathcal{V}}$ is symmetric and positive definite, so there exists an orthogonal V and a diagonal F such that

$$\overline{\mathcal{V}} = VF^{-1}V^\top.$$

Now set $T = V^\top L^{-1/2} U^\top$. Then

$$\overline{\mathcal{V}}_{\tau_1} = T^{-\top} \mathcal{V}_{\tau_1} T^{-1} = I$$

and

$$\overline{\mathcal{V}}_{\tau_2} = T^{-\top} \mathcal{V}_{\tau_2} T^{-1} = F.$$

8.A.3 Proof of Lemma 8.5

By direct integration, it can be verified that

$$K(\tau, A) = \int_{t_k}^{t_k+\tau} W(t-t_k, A) e^{A(t-t_k)}\, dt.$$

We first note that

$$r(t) = -(x(t) - \hat{x}_{\tau_1}(t)) + (x(t) - \hat{x}_{\tau_2}(t)).$$

From Lemma 8.3, it follows that

$$\begin{aligned}
&\int_{t_k}^{t_k+\tau_j} (x(t) - \hat{x}_{\tau_j}(t))\, dt \\
&= w_k \int_{t_k}^{t_k+\tau_j} e^{A(t-t_k)} B dt - w_k \mathcal{V}_{\tau_j}^{-1} W^\top(\tau_j, A) K(\tau_j, A) B \\
&= w_k \left((e^{A\tau_j} - I) A^{-1} - \mathcal{V}_{\tau_j}^{-1} W^\top(\tau_j, A) K(\tau_j, A) \right) B.
\end{aligned}$$

Inserting the equation above into

$$\int_{t_k}^{t_k+\tau_2} r(t)\, dt = -\int_{t_k}^{t_k+\tau_1} (x(t) - \hat{x}_{\tau_1}(t))\, dt + \int_{t_k}^{t_k+\tau_2} (x(t) - \hat{x}_{\tau_2}(t))\, dt$$

proves the lemma.

8.A.4 Proof of Lemma 8.8

The relation in (8.20) follows from

$$\varepsilon(t_o + \tau) = e^{D\tau}\varepsilon(t_o) + w_k e^{D(t_o+\tau-t_k)} B - \hat{w}_l e^{D(t_o+\tau-\hat{t}_l)} B.$$

To prove the last part of the proposition assume, without loss of generality, that $t_o = 0$.

Let $C_i = C(\lambda_i I - D)^{-1}$. The rows of $E(\hat{t})$ are given by

$$\begin{aligned}
E_i(\hat{t}) &= C_i \left(e^{(\lambda_i I - D)\tau} - e^{(\lambda_i I - D)(\tau - \hat{t})} \right) e^{D(\tau-\hat{t})} B \\
&= e^{\lambda_i(\tau-\hat{t})} C_i \left(e^{(\lambda_i I - D)\hat{t}} - I \right) B.
\end{aligned}$$

Note that, for fixed $\hat{t} > 0$,

$$\left| C_i (e^{(\lambda_i I - D)\hat{t}} - I) B \right| \to 0, \text{ as } \lambda_i \to -\infty.$$

Similarly, the rows of $V(\tau - \hat{t}, D)B$ are

$$\begin{aligned} V_i(\hat{t}, D)B &= C_i \left(e^{(\lambda_i I - D)(\tau - \hat{t})} - I \right) e^{D(\tau - \hat{t})} B \\ &= -e^{\lambda_i (\tau - \hat{t})} C_i \left(e^{(D - \lambda_i I)(\tau - \hat{t})} - I \right) B, \end{aligned}$$

Note that, for fixed $0 < \hat{t} < \tau$,

$$\left| C_i (e^{(D - \lambda_i I)(\tau - \hat{t})} - I) B \right| \to \infty, \text{ as } \lambda_i \to -\infty.$$

Hence, it can be concluded that, for $0 < \hat{t} < \tau$,

$$\frac{|E_i(\hat{t})|}{|V_i(\hat{t}, D)B|} = \frac{|C_i (e^{(\lambda_i I - D)\hat{t}} - I) B|}{|C_i (e^{(D - \lambda_i I)(\tau - \hat{t})} - I) B|} \to 0, \text{ as } \lambda_i \to -\infty$$

and thus the result of the lemma follows.

Chapter 9

Identification of the testosterone model

In this chapter, the pulse-modulated Smith model of Section 6.3.3 is extended with infinite-dimensional dynamics, to align it with physiological knowledge and potentially better explain the testosterone concentration profiles observed in clinical data. Furthermore, a method for detecting impulses of GnRH from measured LH concentrations is proposed. Finally, the parameters in the extended model are estimated from hormone concentrations measured in human males, and simulation results from the full pulse-modulated closed-loop system are provided.

9.1 Mathematical model

As shown in Section 6.3.3, the dynamics of the pulse-modulated Smith model (6.6)-(6.8) are thoroughly studied and known to exhibit oscillating solutions that are either periodic or chaotic [169]. In Chapter 7, it was then demonstrated how infinite-dimensional dynamics arising due to e.g. time delays could be analysed using similar mathematical tools.

It is noted in [79] that the LH feedforward on Te is presumably exerted via a time average of the LH concentration. Furthermore, the basal secretion of the hormones should be added to the model, and the time it takes for LH to travel from the pituitary to the testes should be taken into account.

In order to describe this mathematically, we consider the pseudodifferential operator P with symbol

$$p(s) = \frac{1 - e^{-s\ell}}{\ell s} e^{-s\tau}.$$

Clearly, this operator satisfies Assumption 7.1 in Section 7.3, so it is an FM-operator, and for a signal $x(t)$ we get

$$(Px)(t) = \mathcal{L}^{-1}\left\{p(s)X(s)\right\}(t) = \frac{1}{\ell}\int_{t-\tau-\ell}^{t-\tau} x(r)\ \mathrm{d}r, \tag{9.1}$$

i.e., $(Px)(t)$ is a time-delayed average over a sliding window of $x(t)$. According to the discussion above, the LH feedforward on Te should be exerted in this fashion, so we consider the following model:

$$\dot{x}(t) = A_0 x(t) + A_1 (Px)(t) + \beta, \qquad \text{if } t \neq t_k, \tag{9.2}$$

$$x(t_k^+) = x(t_k^-) + w_k B, \qquad \text{if } t = t_k, \tag{9.3}$$

where

$$A_0 = \begin{bmatrix} -b_1 & 0 & 0 \\ g_1 & -b_2 & 0 \\ 0 & 0 & -b_3 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & g_2 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

and the vector $\beta \in \mathbb{R}^3$ describes the basal secretion of the hormones.

Therefore, the basal secretion is seen as a constant signal that is exogenous to the endocrine regulation loop. The impulse times t_k and the weights w_k are still given by (6.8). Note that (9.2) is FD-reducible, see Definition 7.3. The frequency and amplitude modulation functions $\Phi(\cdot)$, and $F(\cdot)$ are here assumed to be Hill functions of the form

$$\begin{aligned} \Phi(y) &= k_1 + k_2 \frac{(y/h)^p}{1 + (y/h)^p}, \\ F(y) &= k_3 + \frac{k_4}{1 + (y/h)^p}, \end{aligned} \tag{9.4}$$

where $p \in \mathbb{N}^+$ is the (positive integer) order of the Hill functions and $k_1, k_2, k_3, k_4, h \in \mathbb{R}^+$.

Remark 9.1. *In [99], the model was also extended with a static nonlinearity that modelled the saturation of the Te release rate for high levels of LH. However, since the resulting model does not fit into the framework of Chapter 7 and the performance gains for including the nonlinearity are relatively small, it is neglected here.*

9.2 Parameter estimation

Given the model defined in Section 9.1, we are interested in estimating the unknown parameters from experimental data. This can be done

using techniques from system identification discussed in Part I of this thesis. In the case of the GnRH-LH-Te axis in the human male, the data usually consist of the LH and Te concentrations measured at certain time instants. The concentration of LH and Te can be assessed from blood samples and are often measured every 10 minutes or similar. GnRH is usually not measured in the human, since it is not possible to do so in a non-invasive way, see, e.g., [78].

The unknown parameters in the model expressed by (9.2)-(9.3) are the constants b_1, b_2, b_3, g_1, and g_2 in the matrices A_0 and A_1. Furthermore, the length of the sliding window ℓ and the time-delay τ in (9.1) have to be estimated.

One problem in identification of the model in (9.2)-(9.3) is, as mentioned above, that the GnRH concentrations are unknown. For this reason, we develop a method for estimating the timing and weights of GnRH impulses in Section 9.2.1. In Section 9.2.2, identification of the parameters describing the Te dynamics is discussed.

9.2.1 Estimating the GnRH impulses

Since the GnRH concentration (9.2) is an unobserved signal in practice, it is of interest to estimate the GnRH impulse times and weights from measured LH data. The main obstacle here is that both the number and the weights of the impulses are unknown. One approach to estimating the pulses was proposed in Chapter 8, but this approach assumes continuous measurements of LH and/or Te concentrations. In practice, the measurements are taken through blood samples, so typically there is at least 10 minutes between each sample. In this situation the methods of Chapter 8 are not reliable. An alternative way for impulse estimation is therefor proposed here.

Before going into details of the estimation procedure, notice first that it is enough to study the first two states of x in (9.2) when the effect of an GnRH impulse on LH is considered. Thus, introduce the reduced (continuous) state vector $\tilde{x}(t) = \begin{bmatrix} x_1(t) & x_2(t) \end{bmatrix}^T$. Also notice that the jump in the state-vector at time t_k, as in (9.3), can equivalently be represented as an input to the system in the form of a delayed Dirac delta impulse, $\delta(t - t_k)$. Therefore, it follows from (9.2)-(9.3) that

$$\dot{\tilde{x}}(t) = \tilde{A}\tilde{x}(t) + \tilde{B}\xi(t) + \tilde{\beta}, \tag{9.5}$$

$$x_2(t) = \tilde{C}\tilde{x}(t), \tag{9.6}$$

where

$$\tilde{A} = \begin{bmatrix} -b_1 & 0 \\ g_1 & -b_2 \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \tilde{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}, \quad \tilde{C} = \begin{bmatrix} 0 & 1 \end{bmatrix}, \tag{9.7}$$

and

$$\xi(t) = \sum_{k=0}^{\infty} w_k \delta(t - t_k). \tag{9.8}$$

Equations (9.5)-(9.6) constitute an LTI-system with the input vector $\tilde{B}\xi(t) + \beta$. If (9.5) is subject to the initial condition $\tilde{x}(t_1) = \tilde{x}_o = \begin{bmatrix} x_{o,1} & x_{o,2} \end{bmatrix}^T$, then the solution to the differential equations in (9.5) is [54]

$$x_2(t) = \tilde{C}\left(\mathrm{e}^{\tilde{A}(t-t_o)}\tilde{x}_o + \int_{t_o}^{t} \mathrm{e}^{\tilde{A}(t-r)}(\tilde{\beta} + \tilde{B}\xi(r))\, \mathrm{d}r \right).$$

By evaluating the integral, the expression above can be rewritten as

$$x_2(t) = \tilde{C}\mathrm{e}^{\tilde{A}(t-t_o)}\tilde{x}_o + \tilde{C}\tilde{A}^{-1}\left(\mathrm{e}^{\tilde{A}(t-t_o)} - I\right)\tilde{\beta} + \sum_{k=0}^{\infty} g_1 w_k z(t-t_k), \tag{9.9}$$

where

$$z(t) = \frac{\mathrm{e}^{-b_2 t} - \mathrm{e}^{-b_1 t}}{b_1 - b_2} H(t),$$

and $H(\cdot)$ is the Heaviside step function.

From (9.9), it can be seen that the unknown parameter g_1 always appears in a product with other unknown parameters, e.g. the impulse weights w_k, cf. [66]. For this reason, it is not possible to uniquely determine g_1 from measured data. Similarly, it is not possible to separate β_1 from β_2 in practice, when the data are collected from the closed-loop system. This intuitively makes sense: Since the basal secretion rate of GnRH is unknown, it is not possible to distinguish between the basal secretion of LH and the LH secreted due to basal GnRH secretion. However, there are evidence that basal GnRH secretion plays no role in the GnRH-LH-Te loop [150], so therefor it is assumed that $\beta_1 = 0$.

Finally, a non-zero initial condition on $\tilde{x}_1(t_1)$ can be replaced by an impulse at time t_1, since there is no way to tell the difference between them based only on measured LH concentrations.

For these reasons, it will be assumed that $\tilde{x}_1(t_1) = 0$, $g_1 = 1$ and $\beta_1 = 0$. Thus, any estimated GnRH signal should be considered as a virtual signal, providing information about the GnRH impulse times t_k and the ratios between the weights w_k. However, the actual concentrations might be different in the real system.

Now assume that we measure the LH concentration at time $\tilde{t}_1, \ldots, \tilde{t}_N$, and stack them in a vector as,

$$Y = \begin{bmatrix} x_2(\tilde{t}_1) & \cdots & x_2(\tilde{t}_N) \end{bmatrix}^\top.$$

Furthermore, let $t_1, \ldots, t_n$ be all the impulse times in the interval $(\tilde{t}_1, \tilde{t}_N)$, i.e. $t_1, \ldots, t_n$ denote all times when a GnRH impulse is secreted from

the hypothalamus during the measurement period. Also let $w_1, \dots, w_n$ be the corresponding impulse weights.

Define

$$\varphi(\tilde{t}_i) = \begin{bmatrix} (1 - e^{-b_2(\tilde{t}_i - \tilde{t}_1)})/b_2 & e^{-b_2(\tilde{t}_i - \tilde{t}_1)} & z(\tilde{t}_i - t_1) & \cdots & z(\tilde{t}_i - t_n) \end{bmatrix}^\top, \tag{9.10}$$

$$\Phi = \begin{bmatrix} \varphi(\tilde{t}_1) & \cdots & \varphi(\tilde{t}_N) \end{bmatrix}^\top, \tag{9.11}$$

$$\theta = \begin{bmatrix} \beta_2 & \tilde{x}_2(\tilde{t}_1) & w_1 & \cdots & w_n \end{bmatrix}^\top. \tag{9.12}$$

It can then easily be verified from (9.9) that

$$Y = \Phi\theta, \tag{9.13}$$

where we have assumed $g_1 = 1$ and $\beta_1 = 0$, as discussed above. However, since Y consists of measured data, it is typically corrupted by disturbances or subject to uncertainty, so that $Y - \Phi\theta \neq 0$. As discussed in Section 2.3.1, these errors are often modeled as a independent stochastic variables with a Gaussian distribution. And as seen in Section 2.4.2, maximizing the likelihood of Y would then be equivalent to the following optimization problem

$$\min_\theta \ \|Y - \Phi\theta\|_2^2. \tag{9.14}$$

Note that, if the only unknown parameters were the ones contained in θ, then this would be just a linear least squares problem which is solvable using techniques discussed in Section 2.5.2. However, in this case, the problem is much harder, since Φ also includes the unknown parameters b_1, b_2 as well as the impulse times $t_1, \dots, t_n$. In fact, we do not even know the number of GnRH impulses n, and thus the dimension of θ is unknown.

Estimating the impulse times and weights

In order to simplify the problem a little, we assume for now that b_1 and b_2 are known. We still have the problem of deciding the number of impulses and their timing. One way to turn (9.14) into a convex optimization problem is to choose a grid of possible impulse times, and insert these into (9.10). That is, instead of using the true, but unknown, impulse times in (9.10) we use the grid points $\hat{t}_1, \dots, \hat{t}_\ell$, and let

$$\widehat{\varphi}(\tilde{t}_i) = \begin{bmatrix} (1 - e^{-b_2(\tilde{t}_i - \tilde{t}_1)})/b_2 & e^{-b_2(\tilde{t}_i - \tilde{t}_1)} & z(\tilde{t}_i - \hat{t}_1) & \cdots & z(\tilde{t}_i - \hat{t}_n) \end{bmatrix}^\top, \tag{9.15}$$

$$\widehat{\Phi} = \begin{bmatrix} \widehat{\varphi}(\tilde{t}_1) & \cdots & \widehat{\varphi}(\tilde{t}_N) \end{bmatrix}^\top, \tag{9.16}$$

and estimate θ as

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|Y - \widehat{\Phi}\theta\|_2^2. \tag{9.17}$$

So how should we construct such a grid? The following lemma suggests that choosing the grid points equal to the sampling times is a good choice.

Lemma 9.1. *Let $\widehat{\Phi}$ be constructed as in* (9.16) *with $\hat{t}_k = \tilde{t}_k$ for $k = 1, \ldots, N-1$, and Φ be given by* (9.11)*. The true, disturbance-free, output is then $\Phi\theta$, as shown in* (9.13)*. The following two statements now hold:*

(i) There is a unique $\hat{\theta}$ such that the first element of $\hat{\theta}$ equals the first element of θ (i.e., $\hat{\beta}_2 = \beta_2$), and $\widehat{\Phi}\hat{\theta} = \Phi\theta$.

(ii) Assume that there are at least three sampling times in between any two impulse times t_k and t_{k+1}. Given the unique $\hat{\theta}$ described in (i), we can uniquely reconstruct θ and the true impulse times $t_1, \ldots, t_n$.

Proof. See Appendix 9.A. □

The above lemma basically states that there is no way, using only sampled measured data, to tell the difference between the noise-free output $\Phi\theta$ and a solution that only has impulse times that coincide with the sampling times.

The lemma can also be used to reconstruct the true impulse times in case the disturbance-free output $\Phi\theta$ as well as the basal secretion β_2 are known. Hence, it seems reasonable to use the sampling times as the grid points in (9.15).

However, in practice, we do not have access to the disturbance-free output, so solving (9.17) will typically add extra impulses in order to describe the disturbance effects. Furthermore, the solution $\hat{\theta}$ is not unique when the basal secretion β_2 is unknown, since the basal secretion can also be described by adding extra impulses. These two properties will typically result in a solution $\hat{\theta}$ that gives an estimated impulse with non-zero weight at every grid point. If the data are, for instance, sampled every 10 minutes, this is not biologically reasonable since there is usually more than one hour between two GnRH impulses in the true system, see e.g. [76]. Another biologically motivated constraint on the problem is that the impulse weights, basal secretion, and initial conditions should all be positive, since concentrations cannot be negative.

We can easily add a non-negativity constraint to (9.17). As hinted above, we should also use some kind of regularization technique to keep the number of non-zero impulse weights low. In order to end up with a convex optimization problem, we enforce this by adding a constraint on the sum of the (positive) impulse weights, that is the ℓ_1-norm, cf.

Section 2.4.3. Hence, we get the following optimization problem:

$$\begin{aligned} \underset{\hat{\theta}}{\operatorname{argmin}} \;\; & \|Y - \hat{\Phi}\hat{\theta}\|_2^2 \\ \text{s.t.} \; & \sum_{k=1}^{N-1} w_k < w_{\max} \\ & \hat{\theta} \geq 0. \end{aligned} \tag{9.18}$$

The impulse estimation scheme can now be summarized as in Algorithm 6. The last part of the algorithm reduces the number of impulses by combining any two consecutive impulses into one impulse, but, as shown in the proof of Lemma 9.1, without affecting the model output at the sampling times.

Note that there might be an ambiguity when three consecutive grid points have been assigned non-zero weights. However, whichever way we resolve this ambiguity in by combining the adjacent impulses, the model output at the sampling times will be the same and one cannot tell which one of them is "better" based only on the data.

Algorithm 6 : **Impulse estimation**

1: Construct $\widehat{\Phi}$ as in (9.16) with $\hat{t}_k = \tilde{t}_k$ for $k = 1, \ldots, N-1$.
2: Solve (9.18).
3: **for all** Two consecutive grid points $\hat{t}_k$ and $\hat{t}_{k+1}$ that have been assigned non-zero impulse weights $\hat{w}_k$ and $\hat{w}_{k+1}$ **do**
4: Let $\Delta = \hat{t}_{k+1} - \hat{t}_k$.
5: Replace the two impulses by one impulse fired at

$$\hat{t} = \hat{t}_k + \frac{1}{b_2 - b_1} \ln\left(1 + \frac{\hat{w}_{k+1}(e^{b_2\Delta} - e^{b_1\Delta})}{\hat{w}_k + \hat{w}_{k+1}e^{b_1\Delta}}\right).$$

6: and impulse weight

$$\hat{w} = e^{-b_1(\hat{t}-\hat{t}_k)}(\hat{w}_{k+1}e^{b_1\Delta} + \hat{w}_k).$$

7: **end for**

Estimating the parameters

In Algorithm 6, we assumed that the parameters b_1 and b_2 were known. If we assumed instead that that the impulses and the basal level are known, then (9.5)-(9.6) would be an LTI-system with known input and could be identified as such [135]. However, since we need b_1 and b_2 in order to estimate the impulses, there seems to be a vicious circle. We

here take a pragmatic approach to resolve this issue. According to [79], b_1 and b_2 should satisfy the bounds

$$\begin{aligned} 0.23 \text{ min}^{-1} \leq b_1 \leq 0.69 \text{ min}^{-1}, \\ 0.0087 \text{ min}^{-1} \leq b_2 \leq 0.014 \text{ min}^{-1}. \end{aligned} \tag{9.19}$$

A simple method to pick an estimate of these parameters is thus to choose a fine grid over the intervals in (9.19), and run Algorithm 6 for each combination of the parameters. Finally, the estimates that give the smallest squared prediction error are selected. There are however alternative approaches, such as the Laguerre domain identification technique suggested in [67]. Another possibility could be to use some kind of cyclic minimization as Section 2.5.4, but since the optimization problem considered here is non-convex, we have no guarantee that such an approach would converge towards a global minimum.

9.2.2 Estimating the testosterone dynamics

Finally, the dynamics from LH to Te have to be identified. From (9.2) it follows that

$$\dot{x}_3(t) = -b_3 x_3(t) + g_2(Px_2)(t) + \beta_3. \tag{9.20}$$

Note that if the time-delay τ and the width of the sliding window ℓ are known, then $(Px_2)(t)$ can be computed for all t using the measured LH concentrations. Hence we have a linear system with known input, and the parameters b_3, g_2 and β_3 can be computed using e.g. the least-squares technique discussed in Section 2.4. Finally, we again grid over possible values of ℓ and τ and choose the estimates that give the least squared prediction error.

9.3 Experimental results

The methods developed in this chapter have been tested on LH and Te concentrations measured in 18 healthy human males. The data were collected for 17 hours and sampled every 10 minutes, see [77] for a description of the data and experimental protocol.

In all patients, the LH data were well explained by the model in (9.5) with impulses estimated using Algorithm 6. For each data set, we obtained a sparse set of 4-10 impulses during the 17 hours of measurements. For illustration, the results of two patients are shown in Figure 9.1. To test the identification of testosterone dynamics, the mathematical model (9.20) was simulated with the measured LH as input. It should be noted

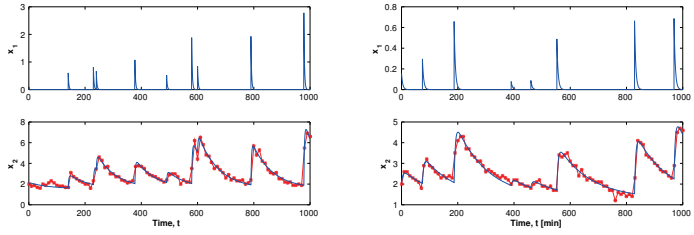


Figure 9.1: Result of GnRH impulses estimation. Upper row shows the estimated virtual GnRH levels for each patient. Bottom row shows the estimated LH levels (solid line), together with the measured data (dashed line with dots at each sampling time). The data in the left column are from a patient that is 27 years old, and in the right column from a 40 years old patient.

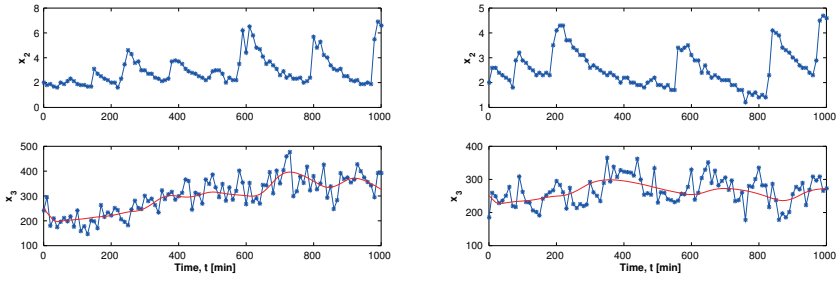


Figure 9.2: Result of Te dynamics identification. The upper row shows the measured LH data that are used as input. The bottom row shows measured Te (blue, dashed line) together with Te concentration given by simulation of the model (red, solid line). The left column corresponds to data from a 27 years old patient, and the right column from a 40 years old patient.

that such a simulation only partly reflects the biological reality since it captures the Te secreted in response to the LH stimulation plus the constant basal secretion. In the actual endocrine system, the concentration of Te is involved in regulations and events outside the GnRH-LH-Te loop, both of endocrine and non-endocrine nature, see e.g. [170] and Chapter 10. However, the simulation results with measured LH data indicate that the simulated Te concentration still follows the general trend in the corresponding measured Te data. For illustration, the result for the same two patients as above are shown in Figure 9.2.

9.4 Simulations of the closed-loop model

In this section, the complete model described by (9.2)-(9.3) is simulated. The model parameters are estimated from the same 27 years old and

40 years old healthy individuals as mentioned in Section 9.3. The modulation functions in (9.4) were then chosen so that the minimum and maximum time between impulses corresponded to the minimum and maximum inter-impulse times in the estimated GnRH concentration.

To imitate the clinical conditions, the simulated data were sampled every 10 minutes with measurement noise added. Figure 9.3-9.4 compare results of the closed-loop simulation with the experimental data that the parameters were estimated from.

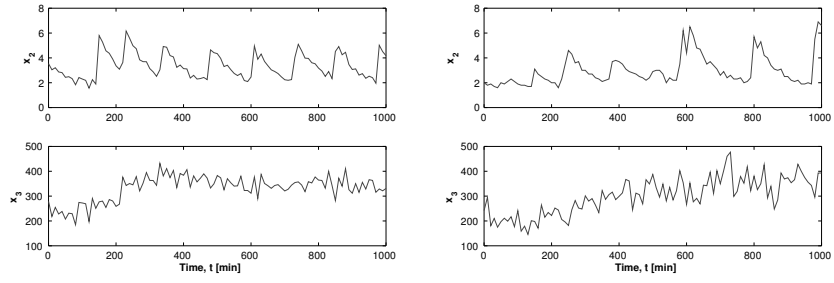


Figure 9.3: Simulation of the closed-loop system. Left column: simulation; right column: measured data in a 27 years old patient.

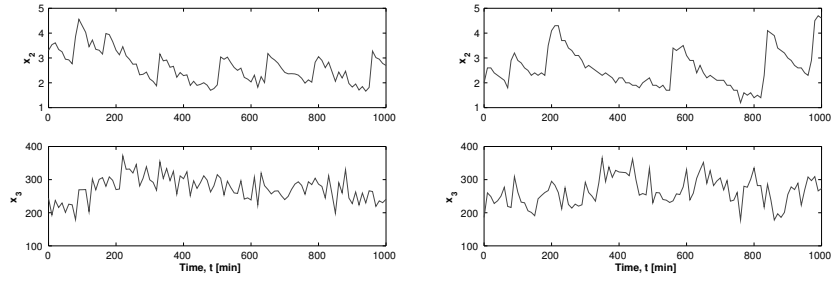


Figure 9.4: Simulation of the closed-loop system. Left column: simulation; right column: measured data in a 40 years old patient.

Since the modulation functions in the feedback are not formally identified from measurements, it can not be expected that the simulated data and the real data would closely resemble each other. However, it can be seen that the simulated and real data in many ways exhibit similarity. For example, the number of impulses are about the same, the hormone concentrations are of the same level, etc. One important difference is that the amplitude of the LH pulses seem to vary more over time in the real data, a phenomenon that can be attributed to a prominent circadian rhythm in Te concentrations [154]. Circadian rhythm can be incorporated by several means in the model, and one suggestion is given

in Chapter 10. However, to make use of such slow periodic behaviors in the identification, one needs data sets that span over several days and nights. Such an experiment is difficult to perform in the human due to limitations on the amount of blood that can be drawn for analysis.

9.A Proof of Lemma 9.1

To start with, note that the first column in Φ and the first column in $\hat{\Phi}$ are the same. Denote $\hat{\Phi} = \begin{bmatrix}[\Phi]_1 & \tilde{\Phi}\end{bmatrix}$, so that $\tilde{\Phi}$ is $\hat{\Phi}$ with the first column removed. Thus row i of $\tilde{\Phi}$ is given by

$$\begin{bmatrix} e^{-b_2(\tilde{t}_i - \tilde{t}_1)} & z(\tilde{t}_i - \tilde{t}_1) & \cdots & z(\tilde{t}_i - \tilde{t}_{N-1}) \end{bmatrix}.$$

Since $z(\tilde{t}_i - \tilde{t}_j)$ is zero if $i \leq j$ and non-zero for $j > i$, it is easy to see that $\tilde{\Phi}$ will be a lower-triangular matrix with the first diagonal element equal to 1, and the ith diagonal element ($i > 1$) given by $z(\tilde{t}_{i+1} - \tilde{t}_i)$. Hence $\tilde{\Phi}$ is invertible, so part (i) of the lemma follows.

Now consider part (ii) of the lemma. In order to prove this, we will first give a constructive way of obtaining $\hat{\theta}$ from θ. If there is a true impulse at time t_i with weight w_i such that $\tilde{t}_j < t_i < \tilde{t}_{i+1}$, then

$$\tilde{x}(\tilde{t}_{j+1}) = e^{\tilde{A}\Delta}\tilde{x}(\tilde{t}_j) + w_i e^{\tilde{A}(\tilde{t}_{j+1} - t_i)}\tilde{B} + \tilde{A}^{-1}\left(e^{\tilde{A}\Delta} - I\right)\tilde{\beta},$$

where $\Delta = \tilde{t}_{j+1} - \tilde{t}_j$. Next let

$$\hat{x}(\tilde{t}_{j+1}^+) = e^{\tilde{A}\Delta}(\tilde{x}(\tilde{t}_j) + \hat{w}_j\tilde{B}) + \hat{w}_{j+1}\tilde{B} + \tilde{A}^{-1}\left(e^{\tilde{A}\Delta} - I\right)\tilde{\beta},$$

so that $\hat{x}(t)$ is the solution we get if we replace the impulse at time t_i with two impulses at time $\tilde{t}_j$ and $\tilde{t}_{j+1}$. Note that if we let

$$\hat{w}_j = w_i \frac{\tilde{C}e^{\tilde{A}(\tilde{t}_{j+1} - t_i)}\tilde{B}}{Ce^{\tilde{A}\Delta}\tilde{B}}, \tag{9.21}$$

then $\hat{x}_2(\tilde{t}_{j+1}) = \tilde{x}_2(\tilde{t}_{j+1})$. Further, if we then let

$$\hat{w}_{j+1} = \begin{bmatrix}1 & 0\end{bmatrix}\left(w_i e^{\tilde{A}(\tilde{t}_{j+1} - t_i)} - \hat{w}_j e^{\tilde{A}\Delta}\right)\tilde{B}, \tag{9.22}$$

then we get $\hat{x}(\tilde{t}_{j+1}^+) = \tilde{x}(\tilde{t}_{j+1})$. That is, we have replaced the impulse at time t_i with two impulses at the sampling times, and this only changes the solution at times $t \in (\tilde{t}_j, \tilde{t}_{j+1})$. By continuing this process for all true impulse times t_i, we thus have a constructive way to obtain $\hat{\theta}$.

From part (i) of the lemma, we know that this $\hat{\theta}$ is unique. So, if we are given $\hat{\theta}$ from part (i), then we have $\hat{w}_j$ that are either zero or

satisfies (9.21)-(9.22). Note that, with the assumption that there is at least three sampling times in between any two impulse times, it follows that there will be at most two consecutive non-zero $\hat{w}_j$ in a row, i.e., if $\hat{w}_j \neq 0$ and $\hat{w}_{j+1} \neq 0$ then $\hat{w}_{j-1} = \hat{w}_{j+2} = 0$.

If it is the case that $\hat{w}_j \neq 0$ and $\hat{w}_{j+1} \neq 0$, then, by the above reasoning, the estimates satisfy (9.21)-(9.22). From these expressions, it can be shown by straightforward but a bit cumbersome calculations that

$$\hat{w}_{j+1}(e^{b_2\Delta} - e^{b_1\Delta}) = w_i \left(e^{b_2(t_i - \tilde{t}_j)} - e^{b_1(t_i - \tilde{t}_j)} \right),$$
$$\hat{w}_{j+1} e^{b_1\Delta} + \hat{w}_j = w_i e^{b_1(t_i - \tilde{t}_j)}.$$

It thus follows that

$$t_i = \tilde{t}_j + \frac{1}{b_2 - b_1} \ln \left(1 + \frac{\hat{w}_{j+1}(e^{b_2\Delta} - e^{b_1\Delta})}{\hat{w}_{j+1} e^{b_1\Delta} + \hat{w}_j} \right), \tag{9.23}$$

$$w_i = e^{-b_1(t_i - \hat{t}_j)} \left(\hat{w}_{j+1} e^{b_1\Delta} + \hat{w}_j \right). \tag{9.24}$$

The expressions in (9.23)-(9.24) thus provide a way to compute the true impulse times and weights given $\hat{\theta}$, so the lemma holds true.

Chapter 10

Modeling of exogenous signals

10.1 Introduction

Closed-loop behaviors of a pulse-modulated testosterone regulation model were studied in Chapter 7. This model was then extended to cover constant basal secretion and identified from clinical data in Chapter 9. While the open-loop dynamics were captured quite well by the model, the closed-loop simulations still exhibited less variation over the day than the clinical data.

In this chapter, we augment the closed-loop dynamics by including an exogenous signal in the model. This exogenous signal can be used to portray basal secretion, either as a constant or with a slowly varying trend, that can capture the circadian rhythm [80]. Furthermore, such an exogenous signal can be used to represent pharmacotherapies of endocrine diseases and conditions, particularly hormone replacement therapies.

Below, we derive a pointwise mapping of the type studied in Chapter 7 that captures the propagation of the continuous states of the model from an impulse time of the pulse-modulated feedback to the next one. It is shown that the effect of the exogenous signal can be equivalently represented by a modification of the frequency and amplitude pulse-modulation functions. As a consequence, the order of the mapping is the same as for the autonomous model thus significantly simplifying the analysis of the complex nonlinear dynamics such as cycles of high multiplicity, quasi-periodic behavior, and chaotic solutions. The results of a bifurcation study of the developed mapping suggest that a constant level of exogenous Te entering the system is most likely to lead to a lower mean Te concentration. Applying a basal Te concentration that varies over the day according to a sinusoidal wave is seen to lead to periodic, quasi-periodic or chaotic solutions.

10.2 The mathematical model

By adding an exogenous signal $\beta(t)$ to the pulse-modulated model in Section 7.5, we get

$$\dot{x}(t) = A_0x(t) + A_1x_\tau(t) + D\beta(t) \qquad \text{if } t \neq t_k, \quad (10.1)$$

$$x(t_k^+) = x(t_k^-) + w_kB_0 \qquad \text{if } t = t_k, \quad (10.2)$$

$$y(t) = Cx(t), \quad (10.3)$$

$$x_\tau(t) = \begin{cases} \varphi(t) & \text{if } t \leq \tau \\ (Px)(t) & \text{if } t > \tau \end{cases} \quad (10.4)$$

$$t_{k+1} = t_k + T_k, \quad T_k = \Phi(y(t_k)), \quad w_k = F(y(t_k)), \quad (10.5)$$

where P is, once again, a pseudodifferential operator with the symbol $p(s)$ that satisfies Assumption 7.1 and has the memory length τ.

We also assume that the model satisfies Assumption 7.2, so that it is FD-reducible, and that

$$A_1A_0^kD = 0, \quad k = 0, 1, \ldots, n_x - 1. \quad (10.6)$$

As we saw in Section 7.4, a system possessing the cascade structure in Figure 7.2 always satisfies Assumption 7.2, and for such a system assumption (10.6) is satisfied when the input $\beta(t)$ only affects the last block.

Lemma 10.1. *The solution $x(t)$ to (10.1)-(10.5) can be written as*

$$x(t) = x_p(t) + z(t),$$

where $x_p(t)$ is governed by

$$\begin{aligned} \dot{x}_p(t) &= A_0x_p(t) + A_1x_\tau(t), \\ x_p(t_k^+) &= x_p(t_k^-) + w_kB_0, \\ x_\tau(t) &= \begin{cases} \varphi(t) & \text{if } t \leq \tau \\ (Px_p)(t) & \text{if } t > \tau \end{cases} \\ t_{k+1} &= t_k + T_k, \quad T_k = \Phi(Cx_p(t_k^-) + Cz(t_k^-)), \\ w_k &= F(Cx_p(t_k^-) + Cz(t_k^-)), \end{aligned}$$

and $z(t)$ satisfies

$$\dot{z}(t) = A_0z(t) + D\beta(t). \quad (10.7)$$

Proof. First note that if $A_1A_0^kD = 0$ for $k = 0, 1, \ldots, n_x - 1$, then $A_1e^{A_0t}D = 0$ for all t. From case (ii) in Lemma 7.3 we also see that

$$A_1e^{A_0t}A_1 = 0, \quad \forall t.$$

By the same reasoning as in the proof of Lemma 7.4, we can then see that $A_1(Px)(t) = A_1(Px_p)(t)$. Hence this lemma follows by noting that

$$\dot{x}_p(t) + \dot{z}(t) = A_0(x_p(t) + z(t)) + A_1 x_\tau(t) + D\beta(t),$$
$$x_p(t_k^+) + z(t_k) = x_p(t_k^-) + z(t_k) + w_k B_0.$$

□

10.3 Pointwise mapping

For the pulse-modulated models in Chapter 7, we saw that pointwise Poincaré mappings were useful in reducing the analysis of the original hybrid dynamical system to a discrete one. The equations governing $x_p(t)$ have the same form as the system studied in Section 7.5, with the only difference being that the modulation function in Lemma 10.1 depends on $z(t)$. Generally, since the evolution of $z(t)$ is independent of $x_p(t)$, the dependence of the modulation functions on $z(t)$ can equivalently be seen as dependence on t_k, i.e. the modulation functions can be seen as time-varying.

Let $x_k = x(t_k^-)$, $x_{p_k} = x_p(t_k^-)$. Then it follows from Lemma 10.1 and Corollary 7.2 that for $k \geq 1$

$$x_{p_{k+1}} = Q_p(x_{p_k}, t_k),$$

where

$$Q_p(x, r) = e^{A\Phi(Cx+Cz(r))}(x + F(Cx + Cz(r))B), \tag{10.8}$$

$A = A_0 + A_1 p(A_0)$, and $B = p(A)p^{-1}(A_0)B_0$. Furthermore, if the initial function is chosen in accordance with (7.23), then the above also holds for $k = 0$. Since $x_k = x_{p_k} + z(t_k)$, the corresponding mapping for $x(t)$ is given by

$$Q(x, r) = Q_p(x - z(r), r) + z(r).$$

Due to the time-varying exogenous signal $\beta(t)$, the mapping now depends on absolute time.

10.3.1 Constant exogenous signal

In the simplest case, the exogenous signal is constant, i.e.

$$\beta(t) = \beta.$$

Assume that A_0 is nonsingular and let $z(0) = -A_0^{-1}D\beta$ so that

$$z(t) = -A_0^{-1}D\beta,$$

for all $t \geq 0$. Notice that the eigenvalues of A_0 are defined by the half-life times of the involved hormones and a zero eigenvalue would mean hormone molecules that do not clear out, which is not biologically reasonable. Hence, assuming that A_0 is invertible is not very restrictive in the case studied here.

Here, the mapping in (10.8) does not depend explicitly on time anymore, since $z(r)$ is a constant. Hence, the only effect a constant exogenous signal has on the mapping is that the modulation functions are shifted by a constant. This also implies that no new complex nonlinear dynamical phenomena can arise due to constant exogenous signal in the pulse-modulated model. Further, the constant shift in the argument of the modulation functions implies that the range of the function (also known as modulation depth or index) is reduced and thus the repertoire of solutions to the closed-loop equations is more limited, compared to the autonomous case.

10.3.2 Alternative formulation

The pointwise mapping in (10.8) is of the same order as the autonomous model in Section 7.5. This is a useful feature since the equivalent discrete dynamics keep the original order no matter what the dynamical complexity of the exogenous signal is. A computational price to pay for this is that the discrete dynamics become time-varying. An alternative approach preserving time-invariance of the pointwise mapping is to augment the continuous state vector of the system, and thus increase the dimension of the mapping.

Consider for example a shifted sinusoidal exogenous signal $\beta(t)$. Such a signal can be generated by the following state-space model

$$\begin{aligned} \dot{w}(t) &= Fw(t), & \beta(t) &= Gw(t), \\ F &= \begin{bmatrix} 0 & -\omega & 0 \\ \omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & G &= \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}, \\ w(0) &= \begin{bmatrix} M\cos\phi \\ M\sin\phi \\ N \end{bmatrix}. \end{aligned}$$

Then

$$\beta(t) = M\sin(\omega t + \phi) + N \tag{10.9}$$

and with a state vector augmentation

$$\bar{x}(t) = \begin{bmatrix} \bar{x}^\top(t) & w^\top(t) \end{bmatrix}^\top,$$

we have

$$\begin{aligned}
\bar{x}(t) &= \bar{A}_0\bar{x}(t) + \bar{A}_1\bar{x}_\tau(t),\\
\bar{x}(t_k^+) &= \bar{x}(t_k^+) + w_k\bar{B}_0,\\
\bar{x}_\tau(t) &= \begin{cases} \bar{\varphi}(t) & \text{if } t \le \tau,\\ (P\bar{x})(t) & \text{if } t > \tau, \end{cases}\\
t_k &= t_k + \Phi(\bar{C}\bar{x}(t)), \quad w_k = F(\bar{C}\bar{x}(t)),
\end{aligned}$$

where

$$\bar{A}_0 = \begin{bmatrix} A_0 & DG \\ 0 & F \end{bmatrix}, \quad \bar{A}_1 = \begin{bmatrix} A_1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \bar{B}_0 = \begin{bmatrix} B_0 \\ 0 \end{bmatrix}, \quad \bar{C} = \begin{bmatrix} C & 0 \end{bmatrix},$$

and $\bar{\varphi}(t) = \begin{bmatrix} \varphi^\top(t) & 0 \end{bmatrix}^\top$. Note that, if the original autonomous model satisfies Assumption 7.2, then this is also true for the extended model. Hence we can, in the same way as in Section 7.5.2, see that the extended state vector has the corresponding mapping

$$\bar{Q}(\bar{x}) = e^{\bar{A}\Phi(\bar{C}x)}(\bar{x} + F(\bar{C}\bar{x})\bar{B}), \tag{10.10}$$

where $\bar{A} = \bar{A}_0 + \bar{A}_1 p(\bar{A}_0)$ and $\bar{B} = p(\bar{A})p^{-1}(\bar{A}_0)\bar{B}_0$.

The above reasoning could easily be generalized to more complicated periodic signal forms by extending the state vector with auxiliary states corresponding to the Fourier series of the exogenous signal. Hence, in this way, a time-invariant mapping can be created even when the exogenous signal is time-varying, with the drawback that the state vector has to be enlarged.

10.4 Periodic solutions

In Section 7.2.1, we discussed how periodic solutions to the finite dimensional pulse-modulated model could be analyzed. It was shown in Section 7.5.2 that this type of analysis can also be used when infinite-dimensional dynamics have been introduced in the closed loop via a finite-memory operator. In this chapter, we have demonstrated that, with a constant exogenous signal, the same type of mapping can be utilized and the analysis in Section 7.2.1 still applies.

However, for general time-varying exogenous signals, the pointwise mapping given in (10.8) is time-varying. So, for example, the equality $x_p(t_{k+1}^-) = Q_p(x_p(t_k^-), t_k) = x_p(t_k^-)$ does not necessarily imply that we have a 1-cycle in this case, since it might happen that $Q(x_p(t_k^-), t_k) \neq Q(x_p(t_k^-), t_{k+1})$.

In general, the propagation of the hybrid state from one impulse time to the next one is governed by

$$\begin{bmatrix} x_p(t_{k+1}^-) \\ t_{k+1} \end{bmatrix} = \begin{bmatrix} Q_p(x_p(t_k^-), g(t_k)) \\ t_k + \Phi(Cx_p(t_k^-) + Cz(t_k)) \end{bmatrix}.$$

Note that t_k strictly increases, and thus the above mapping is not useful in the analysis of periodic solutions.

However, if $\beta(t)$ is periodic then $z(t)$ is periodic, too. In this case, we do not have to keep track of the absolute time t_k, since we only need to know where within the period of $z(t)$ the system is. In a way, this is accomplished by adding $w(t)$ to the state in the extended mapping of (10.10). Another option is to explicitly store the information about where in the period the system is.

Let T_β be the period of $z(t)$, i.e. $z(r) = z(r + T_\beta)$ for all $r \geq 0$. Introduce $g : \mathbb{R}_+ \to [0, T_\beta)$ as

$$g(t) = \min\{\tilde{t} \geq 0 \,|\, \tilde{t} = t - nT_\beta, \text{ for some } n \in \mathbb{N}\},$$

defining the modulo operator, i.e., $g(t) = \mathrm{mod}_{T_\beta}(t)$. Then for all t, one has $z(t) = z(g(t))$. Let $\tilde{x}_{p_k} = \begin{bmatrix} x_p^\top(t_k^-) & g(t_k) \end{bmatrix}$, then

$$\tilde{x}_{p_k} = \tilde{Q}(\tilde{x}_{p_k}),$$

where

$$\tilde{Q}(\tilde{x}_{p_k}) = \begin{bmatrix} Q_p(x_{p_k}, t_k) \\ g(t_k + \Phi(Cx_{p_k} + z(t_k))) \end{bmatrix}.$$

Hence we can now make use of the map $\tilde{Q}(\tilde{x})$ in finding periodic solutions of $x(t)$. Stability can be analysed by checking the eigenvalues of the Jacobian, as in Section 7.2. However, if an m-cycle is considered, we here have to ensure that $g(t)$ has no discontinuity at the impulse times. This is however not very restrictive since $g(t)$ is not part of the actual system, and the discontinuities can thus always be moved by shifting the function.

The lower dimension of this pointwise mapping compared to the extended mapping $\bar{Q}(\bar{x})$ translates into better accuracy of bifurcation analysis. Mappings of high dimension are difficult to study in quasi-periodic, chaotic, and high-multiplicity periodic solutions due to accumulation of numerical errors.

10.5 Numerical examples

To illustrate the complex nonlinear dynamics arising due to the exogenous signal $\beta(t)$, numerical analysis of the system is performed in this section.

We here consider the case when $(Px)(t) = x(t)$, so $A = A_0 + A_1$ and $B = B_0$. We let (10.1)-(10.2) be on the form of the pulse-modulate Smith model of Te-regulation in Section 6.3.3, i.e.,

$$A = \begin{bmatrix} -b_1 & 0 & 0 \\ g_1 & -b_2 & 0 \\ 0 & g_2 & -b_3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$

The modulation functions are, as in previous chapters, chosen as Hill functions,

$$\Phi(y) = k_1 + k_2 \frac{(y/h)^p}{1 + (y/h)^p}, \quad F(y) = k_3 + \frac{k_4}{1 + (y/h)^p}.$$

According to biological data provided in [79], the following parameter intervals apply $0.23\text{min}^{-1} \leq b_1 \leq 0.69\text{min}^{-1}$, and $0.0087\text{min}^{-1} \leq b_2 \leq 0.014\text{min}^{-1}$. In the examples below, b_1 is varied over these values while $b_2 = 0.014$. For the remaining parameters we let $b_3 = 0.15$, $g_2 = 1.5$, $k_1 = 40$, $k_2 = 80$, $k_3 = 0.05$, $k_4 = 5$, $h = 2.7$. Different values of g_1 and p where used in order to highlight different types of behavior in the system.

10.5.1 Constant exogenous Te

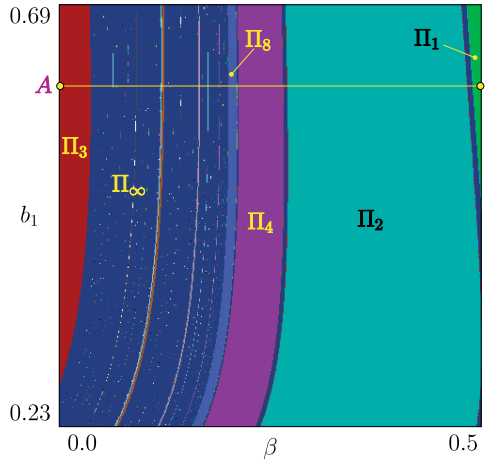


Figure 10.1: Bifurcation structure in the parameter plane (β, b_1) with $b_2 = 0.014$, $b_3 = 0.15$, $g_1 = 2.0$, $g_2 = 1.5$, $p = 4$. Π_i, $i = 1, \ldots, 8$ are the stability domains of i-cycles. Π_∞ is the region of chaotic dynamics.

The exogenous signal is chosen here as a constant $0 \leq \beta \leq 0.5$. An overview of the bifurcation structure in the parameter plane (β, b_1) is provided in Figure 10.1. Here Π_m, $m = 1, \ldots, 8$ are regions with stable

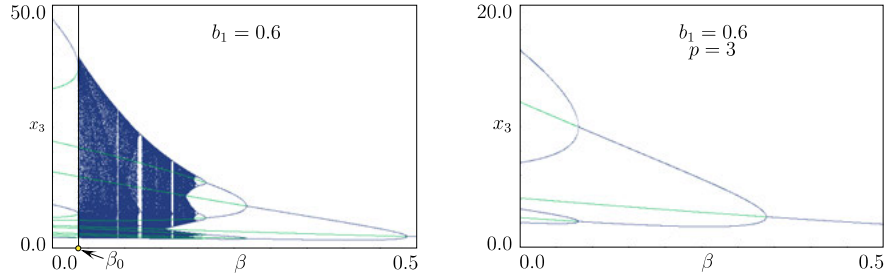


Figure 10.2: Period-doubling route to chaos. The curves of saddle cycles are in green. (a) Bifurcation diagram along A in Figure 10.1 for $b_1 = 0.6$, $g_1 = 2.0$, $p = 4$; β_0 is the saddle-node bifurcation point. (b) Finite sequence of period-doubling bifurcations for $b_1 = 0.6$, $g_1 = 0.6$, and $p = 3$.

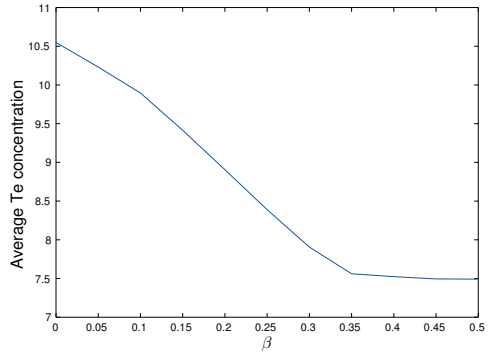


Figure 10.3: Average total Te concentration as a function of β = const; $b_1 = 0.6$, $g_1 = 0.6$, and $p = 3$

m-cycles. The domain of stability for Π_3 is bounded from the right by a saddle-node bifurcation curve. The domains Π_{2^i}, $i = 0, 1, 2, \ldots$ of the stable dynamics are separated by period-doubling bifurcation curves. Transverse to these curves are the curves along which the accumulating period-doubling cascades occur. The dark blue region Π_∞ is a region of chaotic dynamics with a usual dense set of periodic windows.

Figure 10.2(a) displays a one-dimensional bifurcation diagram along the scan line A in Figure 10.1. With decreasing β, an infinite cascade of period-doubling bifurcations is observed, resulting in a transition to chaos. As the value of β continues to decrease, the system enters a period-3 window Π_3 through a saddle-node bifurcation at β_0. Figure 10.2(b) illustrates an example of a finite sequence of period-doubling bifurcations for $g_1 = 0.6$ and $p = 3$. The green drawn curves

in Figure 10.2(a),(b) represent saddle cycles. Apparently, higher values of β lead to less oscillative hormonal variation pattern.

Due to the negative feedback mechanism, increasing the exogenous Te decreases the total average Te concentration, see Figure 10.3. Thus, according to the model, it is impossible to elevate the mean Te concentration in the male by administering a constant influx of exogenous Te without saturating the pulse-modulated feedback. Indeed, adverse effects of Te replacement therapies are now well known [117].

10.5.2 Periodic exogenous Te

In this section, the parameters are selected as $b_1 = 0.55$, $g_1 = 0.6$, $p = 3$. The exogenous signal is chosen as in (10.9), with $\omega = 2\pi/T$, $T = 1440$ min and $N = M$.

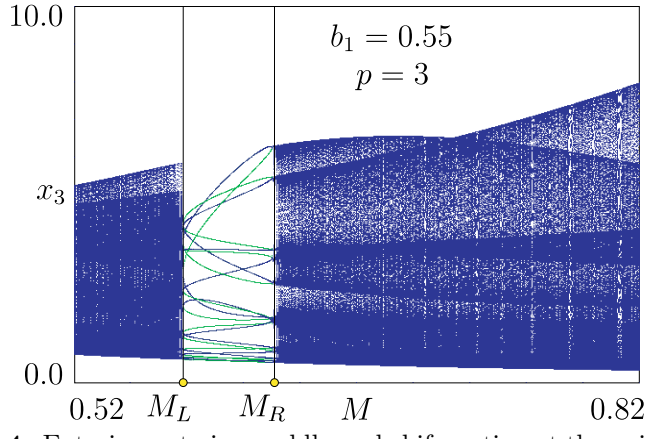
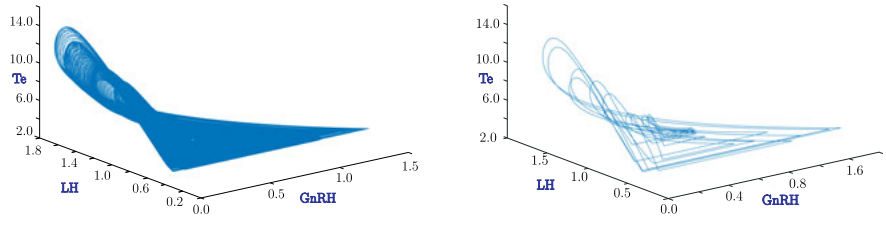


Figure 10.4: Entrainment via a saddle-node bifurcation at the points M_L and M_R. The blue lines represent a stable 13-cycle and green lines mark a saddle one. $\beta(t)$ is given by (10.9), $N = M$, $b_1 = 0.55$, $g_1 = 0.6$, and $p = 3$.

Figure 10.4 displays the bifurcation diagram for $0.52 < M < 0.82$ illustrating the mechanism of entrainment. The autonomous system ($M = 0$) exhibits a stable 8-cycle. As M is increased from zero, the system dynamics become quasiperiodic. With further increase of M, the system enters the region of entrainment via a saddle-node bifurcation at $M = M_L$. Under entrainment ($M_L \leqslant M \leqslant M_R$), the mapping has a pair of 13-cycles (i.e. 13 pulses in $T = 24$ hours), one of which is stable (in blue), while the other is a saddle (in green), lying on a stable closed invariant curve. This attracting set is formed by the unstable manifolds of the saddle 13-cycle and the points of the corresponding saddle and stable 13-cycles. When M increases or decreases, the stable and unstable cycles collide and disappear in a fold (saddle-node) bifurcation at

the points M_R and M_L, respectively. Phase portraits of a quasiperiodic attractor and a stable 13-cycle are shown in Figure 10.5.



(a) **(b)**

Figure 10.5: Phase portrait of the solutions under periodic exogenous Te: (a) – quasiperiodic attractor, (b) – stable 13-cycle.

Concluding remarks

This thesis covers a number of topics in modeling of nonlinear and impulsive systems. Already in Chapter 1 different strategies to modeling were discussed, and the importance of considering the purpose of the model was emphasized.

In Part I of this thesis, black-box system identification of nonlinear systems was considered. Since, as noted in Chapter 1, system nonlinearity is not a limiting assumption, we opted for model structures that can express a wide range of dynamical behaviors.

In Chapter 3, the LAVA-framework for identifying predictor models was presented. Even though the framework is aimed towards prediction, it was seen in the numerical example that the identified models typically perform well in simulations too. In a nutshell, the proposed method starts from a nominal model and then adds nonlinear elements, if the nominal prediction errors indicate that this is beneficial. In order to stay true to the principle of parsimony, the method is biased towards the nominal model, and it also avoids the tuning of regularization parameters by the user. In principal, any NARX model structure based on basis expansions can be expressed within the LAVA-framework, and thus it allows a wide range of different dynamical behaviors. In particular, it was shown in Chapter 4 that piecewise ARX-models can be identified using LAVA.

In Chapter 5, the popular Hammerstein models were studied, and a recursive identification algorithm was derived. Furthermore, a convergence analysis gave rise to both theoretical insights into the method as well as practical insights into the parameterization of the model structure.

In Part II, the purpose was to find a model that describes the behavior of testosterone regulation in the human male. Since this endocrine subsystem has been intensively studied in the literature, the model was based on biological knowledge, and available clinical data then made it

possible to estimate the unknown parameters. In order to analyze the model at hand, new results on the dynamics of the models with intrinsic pulse-modulated feedback were obtained in Chapter 7, and methods for estimating the immeasurable hormone concentrations were developed in Chapter 8. The constructed model was then summarized in Chapter 9, together with a method for identifying the unknown parameters. It was seen that the luteinizing hormone profiles could be explained very well by the model, but that the testosterone concentrations exhibit slow variations that are not caputred by the model, presumably due to the circadian rhythm. Finally, in Chapter 10, the effect of exogenous signals on the model was considered. These signals can be used in order to describe the effects due to the circadian rhythm, or medical interventions.

References

[1] K. J. Åström and P. Kumar. Control: A perspective. *Automatica*, 50(1):3–43, 2014.

[2] R. Abraham, H. Kocak, and W. Smith. Chaos and intermittency in an endocrine system model. In P. Fischer and W. R. Smith, editors, *Chaos, Fractals, and Dynamics*, volume 98, pages 33 – 70. Marcel Dekker, Inc., 1985.

[3] P. Alper. A consideration of the discrete Volterra series. *IEEE Transactions on Automatic Control*, 10(3):322–327, 1965.

[4] V. Anishchenko. Dynamical chaos–models and experiments. *World Scientific Series on Nonlinear Science–Series A*, 8, 1995.

[5] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[6] E.-W. Bai and D. Li. Convergence of the iterative Hammerstein system identification algorithm. *IEEE Transactions on Automatic Control*, 49(11):1929–1940, Nov 2004.

[7] E.-W. Bai and Y. Liu. Recursive direct weight optimization in nonlinear system identification: A minimal probability approach. *IEEE Transactions on Automatic Control*, 52(7):1218–1231, 2007.

[8] L. Bako, K. Boukharouba, E. Duviella, and S. Lecoeuche. A recursive identification algorithm for switched linear/affine models. *Nonlinear Analysis: Hybrid Systems*, 5(2):242 – 253, 2011.

[9] D. Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[10] M. Bask and A. Medvedev. Analysis of least-squares state estimators for a harmonic oscillator. In *Proceedings of the 39th IEEE Conference on Decision and Control, 2000.*, volume 4, pages 3819–3824 vol.4, 2000.

[11] R. M. Bell, Y. Koren, and C. Volinsky. All together now: a perspective on the Netflix prize. *Chance*, 23(1):24–29, 2010.

[12] A. Belloni, V. Chernozhukov, and L. Wang. Square-root LASSO: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.

[13] A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control*, 45(10):1864–1876, 2000.

[14] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50(10):1567–1580, Oct 2005.

[15] H. Bijl, J.-W. van Wingerden, T. B. Schön, and M. Verhaegen. Online sparse Gaussian process regression using FITC and PITC approximations. *IFAC-PapersOnLine*, 48(28):703–708, 2015.

[16] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[17] M. Boutayeb, D. Aubry, and M. Darouach. A robust and recursive identification method for MISO Hammerstein model. In *Control '96, UKACC International Conference on (Conf. Publ. No. 427)*, volume 1, pages 234–239 vol.1, Sept 1996.

[18] G. E. Box and N. R. Draper. *Empirical model-building and response surfaces*, volume 424. Wiley New York, 1987.

[19] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.

[20] S. Boyd and L. Chua. Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on Circuits and Systems*, 32(11):1150–1161, 1985.

[21] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[22] L. Breiman. Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transansactions on Information Theory*, 39(3):999–1013, May 1993.

[23] F. Chang and R. Luus. A noniterative method for identification using Hammerstein model. *IEEE Transactions on Automatic Control*, 16(5):464–468, 1971.

[24] H. Chen. Extended recursive least squares algorithm for nonlinear stochastic systems. In *American control conference, 2004*, volume 5, pages 4758–4763. IEEE, 2004.

[25] H.-F. Chen. Pathwise convergence of recursive identification algorithms for Hammerstein systems. *IEEE Transactions on Automatic Control*, 49(10):1641–1649, 2004.

[26] A. Churilov, A. Medvedev, and P. Mattsson. Analysis of a pulse-modulated model of endocrine regulation with time-delay. In *51st IEEE Conference on Decision and Control*, pages 362–367, 2012.

[27] A. Churilov, A. Medvedev, and P. Mattsson. On Finite-dimensional Reducibility of Time-delay Systems under Pulse-modulated Feedback. In *52nd IEEE Conference on Decision and Control*, pages 362–367, 2013.

[28] A. Churilov, A. Medvedev, and P. Mattsson. Discrete-time modeling of a hereditary impulsive feedback system. In *53rd IEEE Conference on Decision and Control*, Los Angeles, California, USA, Dec. 2014.

[29] A. Churilov, A. Medvedev, and P. Mattsson. Periodical solutions in a pulse-modulated model of endocrine regulation with time-delay. *IEEE Transactions on Automatic Control*, 59(3):728–733, March 2014.

[30] A. Churilov, A. Medvedev, and A. Shepeljavyi. Mathematical model of non-basal testosterone regulation in the male by pulse modulated feedback. *Automatica*, 45(1):78–85, 2009.

[31] A. Churilov, A. Medvedev, and A. Shepeljavyi. Further results on a state observer for continuous oscillating systems under intrinsic pulsatile feedback. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 5443–5448, Dec. 2011.

[32] A. Churilov, A. Medvedev, and A. Shepeljavyi. A state observer for continuous oscillating systems under intrinsic pulse-modulated feedback. *Automatica*, 48(6):1117–1122, 2012.

[33] A. N. Churilov and A. Medvedev. Discrete-time map for an impulsive Goodwin oscillator with a distributed delay. *Mathematics of Control, Signals, and Systems*, 28(1):1–22, 2016.

[34] F. Clément and J.-P. Françoise. Mathematical modeling of the GnRH-pulse and surge generator. *SIAM J. Appl. Dynam. Syst.*, 6(2):441–456, 2007.

[35] G. De Nicolao and D. Liberati. Linear and nonlinear techniques for the deconvolution of hormone time-series. *IEEE Transactions on Biomedical Engineering*, 40(5):440–455, May 1993.

[36] L. Debnath and D. Bhatta. *Integral transforms and their applications*. CRC Press, second edition, 2007.

[37] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, pages 1–38, 1977.

[38] D. Dierschke, A. Bhattacharya, L. Atkinson, and E. Knobil. Circhoral oscillations of plasma LH levels in the ovariectomized rhesus monkey. *Endocrinology*, 87:850–853, 1970.

[39] F. Ding, X. Liu, and G. Liu. Identification methods for Hammerstein nonlinear systems. *Digital Signal Processing*, 21(2):215 – 238, 2011.

[40] F. Ding, L. Xinggao, and J. Chu. Gradient-based and least-squares-based iterative algorithms for Hammerstein systems using the hierarchical identification principle. *IET Control Theory & Applications*, 7:176–184, 2013.

[41] D. V. Efimov and A. L. Fradkov. Oscillatority conditions for nonlinear systems with delay. *Journal of Applied Mathematics*, 2007.

[42] Y. V. Egorov. *Linear differential equations of principal type*. Consultants bureau, New York, 1986.

[43] G. Enciso and E. Sontag. On the stability of a model of testosterone dynamics. *J. Math. Biol.*, 49:627–634, 2004.

[44] M. Enqvist. *Linear models of nonlinear systems*. PhD thesis, Linköping University, Linköping, Sweden, 2005.

[45] W. S. Evans, L. S. Farhy, and M. L. Johnson. Biomathematical modeling of pulsatile hormone secretion: a historical perspective. *Methods in enzymology*, 454:345–366, 2009.

[46] T. Falck, J. A. Suykens, J. Schoukens, and B. De Moor. Nuclear norm regularization for overparametrized hammerstein systems. In *49th IEEE Conference on Decision and Control*, pages 7202–7207, 2010.

[47] L. S. Farhy. Modeling of oscillations in endocrine networks with feedback. In M. L. Johnson and L. Brand, editors, *Numerical Computer Methods, Part E*, volume 384 of *Methods in Enzymology*, pages 54–81. Academic Press, 2004.

[48] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205 – 217, 2003.

[49] J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[50] A. Garulli, S. Paoletti, and A. Vicino. A survey on switched and piecewise affine system identification. In *16th IFAC Symposium on System Identification*, volume 16, pages 344–355, 2012.

[51] C. F. Gauss. Theoria motus corporum celestium. hamburg: perthes et besser. Translated, 1857, as Theory of motion of the heavenly bodies moving about the sun in conic sections, trans. CH Davis. Reprinted, 1963, 1809.

[52] A. K. Gelig and A. N. Churilov. Frequency methods in the theory of pulse-modulated control systems. *Automation and Remote Control*, 67(11):1752–1767, 2006.

[53] F. Giri and E.-W. Bai. *Block-oriented nonlinear system identification*, volume 1. Springer, 2010.

[54] T. Glad and L. Ljung. *Control Theory : Multivariable and Nonlinear Methods*. Taylor and Francis, New York, London, 2000.

[55] R. Goebel, R. G. Sanfelice, and A. R. Teel. *Hybrid Dynamical Systems: modeling, stability, and robustness*. Princeton University Press, 2012.

[56] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008.

[57] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1, Mar. 2014.

[58] W. Haddad, V. Chellaboina, and S. Nersesov. *Impulsive and Hybrid Dynamical Systems: Stability, Dissipativity, and Control*. Princeton Series in Applied Mathematics. Princeton University Press, 2006.

[59] W. M. Haddad and V. Chellaboina. *Nonlinear dynamical systems and control: a Lyapunov-based approach*. Princeton University Press, 2008.

[60] G. Hamerly and C. Elkan. Alternatives to the k-means algorithm that find better clusterings. In *Proceedings of the 11th International Conference on Information and Knowledge Management*, pages 600–607. ACM, 2002.

[61] A. Hammerstein. Nichtlineare integralgleichungen nebst anwendungen. *Acta Mathematica*, 54(1):117–176, 1930.

[62] Y. Han and R. A. De Callafon. Hammerstein system identification using nuclear norm minimization. *Automatica*, 48(9):2189–2193, 2012.

[63] M. R. Hassan and B. Nath. Stock market forecasting using hidden Markov model: a new approach. In *5th international conference on Intelligent Systems Design and Applications*, pages 192–196. IEEE, 2005.

[64] W. J. Heuett and H. Qian. A stochastic model of oscillatory blood testosterone levels. *Bulletin of Mathematical Biology*, 68(6):1383–1399, 2006.

[65] E. Hidayat. On Identification of Endocrine Systems. IT licentiate theses / Uppsala University, Department of Information Technology, 2012.

[66] E. Hidayat and A. Medvedev. Identification of a pulsatile endocrine model from hormone concentration data. In *IEEE International Conference on Control Applications*, pages 356–363, 2012.

[67] E. Hidayat and A. Medvedev. Laguerre domain identification of continuous linear time-delay systems from impulse response data. *Automatica*, 48(11):2902 – 2907, 2012.

[68] R. A. Horn and C. R. Johnson. *Topics in matrix analysis*. Cambridge Univ. Press Cambridge etc, 1991.

[69] J.D. Murray. *Mathematical Biology, I: An Introduction (3rd ed.)*. Springer, New York, 2002.

[70] W. H. Jefferys and J. O. Berger. Sharpening Ockham's razor on a Bayesian strop. *Dept. Statistics, Purdue Univ., West Lafayette, IN, Tech. Rep*, 1991.

[71] M. L. Johnson, L. Pipes, P. P. Veldhuis, L. S. Farhy, R. Nass, M. O. Thorner, and W. S. Evans. AutoDecon: A robust numerical method for the quantification of pulsatile events. In M. L. Johnson and L. Brand, editors, *Methods in Enzymology: Computer Methods, Volume A*, volume 454, pages 367–404. Elsevier, 2009.

[72] A. Juloski, S. Weiland, and W. Heemels. A Bayesian approach to identification of hybrid systems. *IEEE Transactions on Automatic Control*, 50(10):1520–1533, Oct 2005.

[73] A. L. Juloski, W. Heemels, and G. Ferrari-Trecate. Data-based hybrid modelling of the component placement process in pick-and-place machines. *Control Engineering Practice*, 12(10):1241–1252, 2004.

[74] A. L. Juloski, S. Paoletti, and J. Roll. Recent techniques for the identification of piecewise affine and hybrid systems. In *Current trends in nonlinear systems and control*, pages 79–99. Springer, 2006.

[75] T. Kailath. *Linear systems*, volume 156. Prentice-Hall Englewood Cliffs, NJ, 1980.

[76] D. Keenan, S. Chattopadhyay, and J. Veldhuis. Composite model of time-varying appearance and disappearance of neurohormone pulse signals in blood. *Journal of Theoretical Biology*, 236(3):242–255, 2005.

[77] D. Keenan, I. Clarke, and J. Veldhuis. Non-invasive analytical estimation of endogenous gonadotropin-releasing hormone (GnRH) drive: analysis using graded competitive GnRH-receptor antagonism and a calibrating pulse of exogenous GnRH. *Endocrinology*, 152(12):4882–93, 2011.

[78] D. Keenan, W. Sun, and J. Veldhuis. A stochastic biomathematical model of male reproductive hormone systems. *SIAM J. Appl. Math.*, 61(3):934–965, 2000.

[79] D. M. Keenan and J. D. Veldhuis. A biomathematical model of time-delayed feedback in the human male hypothalamic-pituitary-leydig

cell axis. *American Journal of Physiology-Endocrinology And Metabolism*, 275(1):E157–E176, 1998.

[80] D. M. Keenan and J. D. Veldhuis. Pulsatility of hypothalamo-pituitary hormones: A challenge in quantification. *Physiology*, 31:34–50, 2016.

[81] H. C. Koh, G. Tan, et al. Data mining applications in healthcare. *Journal of healthcare information management*, 19(2):65, 2011.

[82] B. P. Kovatchev, M. Breton, C. Dalla Man, and C. Cobelli. In silico preclinical trials: A proof of concept in closed-loop control of type 1 diabetes. *Journal of Diabetes Science and Technology*, 3(1):44–55, January 2009.

[83] F. Lauer. On the complexity of piecewise affine system identification. *Automatica*, 62:148–153, 2015.

[84] C. L. Lawson and R. J. Hanson. *Solving least squares problems*, volume 15. SIAM, 1995.

[85] J.-N. Lin and R. Unbehauen. Canonical piecewise-linear approximations. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 39(8):697–699, Aug 1992.

[86] L. Ljung. Analysis of recursive stochastic algorithms. *IEEE Transactions on Automatic Control*, 22(4):551–575, Aug 1977.

[87] L. Ljung. Model validation and model error modeling. In *The Åström Symposium on Control*, pages 15–42. Studentlitteratur, 1999.

[88] L. Ljung. *System identification (2nd ed.): theory for the user*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.

[89] L. Ljung. *System identification toolbox for use with MATLAB*. The MathWorks, Inc., 2007.

[90] L. Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1 – 12, 2010.

[91] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. MIT Press, Cambridge, MA, 1983.

[92] L. Ljung and B. Wahlberg. Asymptotic properties of the least-squares method for estimating transfer functions and disturbance spectra. *Advances in Applied Probability*, pages 412–440, 1992.

[93] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[94] D. Luenberger. Observers for multivariable systems. *IEEE Transactions on Automatic Control*, 11(2):190–197, apr 1966.

[95] P. Lynch. The origins of computer weather prediction and climate modeling. *Journal of Computational Physics*, 227(7):3431–3444, 2008.

[96] L. Magni, G. Bastin, and V. Wertz. Multivariable nonlinear predictive control of cement mills. *IEEE Transactions on Control Systems Technology*, 7(4):502–508, Jul 1999.

[97] A. Mattsson. *Roles of ERα and ERβ in normal and disrupted sex differentiation in japanese quail*. PhD thesis, Uppsala University, 2008.

[98] P. Mattsson and A. Medvedev. Estimation of input impulses by means of continuous finite memory observers. In *American Control Conference*, pages 6769–6774, June 2012.

[99] P. Mattsson and A. Medvedev. Modeling of testosterone regulation by pulse-modulated feedback: an experimental data study. *AIP Conference Proceedings*, 1559(1), 2013.

[100] P. Mattsson and A. Medvedev. State estimation in linear time-invariant systems with unknown impulsive inputs. In *European Control Conference*, pages 1675–1680, July 2013.

[101] P. Mattsson and A. Medvedev. Modeling of testosterone regulation by pulse-modulated feedback. In *Signal and Image Analysis for Biomedical and Life Sciences*, pages 23–40. Springer, 2015.

[102] P. Mattsson, A. Medvedev, and A. Churilov. Poincaré map for an Impulsive Oscillator with General Hereditary Dynamics. *Submitted*, 2016.

[103] P. Mattsson, A. Medvedev, and Z. Zhusubaliyev. Pulse-modulated Model of Testosterone Regulation Subject to Exogenous Signals. In *55th IEEE Conference on Decision and Control*, Las Vegas, USA, Dec. 2016.

[104] P. Mattsson and T. Wigren. Recursive identification of Hammerstein models. In *American Control Conference*, June 2014.

[105] P. Mattsson and T. Wigren. Convergence analysis for recursive Hammerstein identification. *Automatica*, 71:179–186, 2016.

[106] P. Mattsson, D. Zachariah, and P. Stoica. Recursive identification method for piecewise ARX models: A sparse estimation approach. *IEEE Transactions on Signal Processing*, 64(19):5082–5093, 2016.

[107] P. Mattsson, D. Zachariah, and P. Stoica. Recursive nonlinear system identification method using latent variables. *Submitted*, 2016.

[108] J. Mayer, K. Khairy, and J. Howard. Drawing an elephant with four complex parameters. *American Journal of Physics*, 78(6):648–649, 2010.

[109] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions.* Wiley, New York, 1997.

[110] E. Medina and D. Lawrence. State estimation for linear impulsive systems. In *American Control Conference*, pages 1183–1188, june 2009.

[111] A. Medvedev. Continuous least-squares observers with applications. *IEEE Transactions on Automatic Control*, 41(10):1530–1537, Oct. 1996.

[112] A. Medvedev. Disturbance attenuation in finite-spectrum-assignment controllers. *Automatica*, 33(6):1163–1168, 1997.

[113] A. Medvedev. State estimation and fault detection by a bank of continuous finite-memory filters. *International Journal of Control*, 69(4):499–517, 1998.

[114] A. Medvedev and H. T. Toivonen. Directional sensitivity of continuous least-squares state estimators. *Systems & Control Letters*, 59(9):571–577, 2010.

[115] H. Nakada, K. Takaba, and T. Katayama. Identification of piecewise affine systems based on statistical clustering technique. *Automatica*, 41(5):905 – 913, 2005.

[116] K. Narendra and P. Gallman. An iterative method for the identification of nonlinear systems using a Hammerstein model. *IEEE Transactions on Automatic Control*, 11(3):546–550, Jul 1966.

[117] E. Nieschlag, H. M. Behre, P. Bouchard, J. J. Corrales, T. H. Jones, G. Stalla, S. Webb, and F. Wu. Testosterone replacement therapy: current trends and future directions. *Human Reproduction Update*, 10(5), 2004.

[118] H. Ohlsson and L. Ljung. Identification of switched linear regression models using sum-of-norms regularization. *Automatica*, 49(4):1045–1050, 2013.

[119] R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The effectiveness of Lloyd-type methods for the k-means problem. In *47th Annual IEEE Symposium on Foundations of Computer Science, 2006*, pages 165–176. IEEE, 2006.

[120] J. Paduart, L. Lauwers, J. Swevers, K. Smolders, J. Schoukens, and R. Pintelon. Identification of nonlinear systems using polynomial nonlinear state space models. *Automatica*, 46(4):647 – 656, 2010.

[121] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal. Identification of hybrid systems a tutorial. *European Journal of Control*, 13(2-3):242 – 260, 2007.

[122] V. Peterka. Bayesian system identification. *Automatica*, 17(1):41–53, 1981.

[123] R. Pintelon and J. Schoukens. *System identification: a frequency domain approach*. John Wiley & Sons, 2012.

[124] T. Raff and F. Allgower. An impulsive observer that estimates the exact state of a linear continuous-time system in predetermined finite time. In *Mediterranean Conference on Control Automation*, pages 1–3, June 2007.

[125] N. Rasgon, L. Pumphrey, P. Prolo, S. Elman, A. Negrao, J. Licinio, and A. Garfinkel. Emergent oscillations in mathematical model of the human menstrual cycle. *CNS Spectrums*, 8(11):805–814, 2003.

[126] J. Roll, A. Bemporad, and L. Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1):37 – 50, 2004.

[127] J. Roll, A. Nazin, and L. Ljung. Nonlinear system identification via direct weight optimization. *Automatica*, 41(3):475–490, 2005.

[128] W. J. Rugh. *Linear system theory*, volume 2. Prentice Hall Upper Saddle River, NJ, 1996.

[129] M. Schetzen. *The Volterra and Wiener theories of nonlinear systems*. Wiley, 1980.

[130] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: A unified overview. *Automatica*, 31(12):1691 – 1724, 1995. Trends in System Identification.

[131] R. S. Smith. Nuclear norm minimization methods for frequency domain subspace identification. In *American Control Conference*, pages 2689–2694. IEEE, 2012.

[132] R. S. Smith and J. C. Doyle. Model validation: a connection between robust control and identification. *IEEE Transactions on Automatic Control*, 37(7):942–952, 1992.

[133] W. R. Smith. Hypothalamic regulation of pituitary secretion of luteinizing hormone. II. feedback control of gonadotropin secretion. *Bulletin of Mathematical Biology*, 42(1):57 – 78, 1980.

[134] T. Söderström. *Discrete-time stochastic systems: estimation and control*. Springer Science & Business Media, 2012.

[135] T. Söderström and P. Stoica. *System Identification*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[136] A. Solin and S. Särkkä. Hilbert space methods for reduced-rank Gaussian process regression, 2014. arXiv preprint arXiv:1401.5508.

[137] E. D. Sontag. Interconnected automata and linear systems: A theoretical framework in discrete-time. In *Hybrid systems III*, pages 436–448. Springer, 1996.

[138] G. Sparacino, G. Pillonetto, M. Capello, G. D. Nicolao, and C. Cobelli. Winstodec: A stochastic deconvolution interactive program for physiological and pharmacokinetic systems. *Computer Methods and Programs in Biomedicine*, 67(1):67 – 77, 2002.

[139] C. Sparrow. Chaos in a three-dimensional single loop feedback system with a piecewise linear feedback function. *J. Math. Anal. Appl.*, 83(1):275–291, 1981.

[140] M. L. Stein. *Interpolation of Spatial data: Some Theory for Kriging*. Springer Science & Business Media, 1999.

[141] P. Stoica and P. Åhgren. Exact initialization of the recursive least-squares algorithm. *International Journal of Adaptive Control and Signal Processing*, 16(3):219–230, 2002.

[142] P. Stoica and R. L. Moses. *Spectral analysis of signals*. Pearson/Prentice Hall Upper Saddle River, NJ, 2005.

[143] P. Stoica and Y. Selén. Cyclic minimizers, majorization techniques, and the expectation-maximization algorithm: A refresher. *IEEE Signal Processing Magazine*, 21(1):112–114, 2004.

[144] P. Stoica and T. Söderström. Instrumental-variable methods for identification of Hammerstein systems. *International Journal of Control*, 35(3):459–476, 1982.

[145] P. Stoica, D. Zachariah, and J. Li. Weighted SPICE: A unifying approach for hyperparameter-free sparse estimation. *Digital Signal Processing*, 33:1–12, 2014.

[146] R. Strichartz. *A Guide to Distribution Theory and Fourier Transforms*. CRC Press, 1994.

[147] S. Tayamon, T. Wigren, and J. Schoukens. Convergence analysis and experiments using an RPEM based on nonlinear ODEs and midpoint integration. In *IEEE 51st Annual Conference on Decision and Control*, pages 2858–2865. IEEE, 2012.

[148] Y. Tian, T. Floquet, L. Belkoura, and W. Perruquetti. Algebraic switching time identification for a class of linear hybrid systems. *Nonlinear Analysis: Hybrid Systems*, 5(2):233–241, 2011.

[149] R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):pp. 267–288, 1996.

[150] R. Tsutsumi and N. J. Webster. GnRH pulsatility, the pituitary response and reproductive dysfunction. *Endocrine Journal*, 56(6):729, 2009.

[151] P. Van den Hof and B. Ninness. System identification with generalized orthonormal basis functions. In *Modelling and Identification with Rational Orthogonal Basis Functions*, pages 61–102. Springer, 2005.

[152] J. Varah. A lower bound for the smallest singular value of a matrix. *Linear Algebra and its Applications*, 11(1):3 – 5, 1975.

[153] J. D. Veldhuis. Recent insights into neuroendocrine mechanisms of aging of the human male hypothalamic-pituitary-gonadal axis. *Journal of Andrology*, 20(1):1–18, 1999.

[154] J. D. Veldhuis, J. C. King, R. J. Urban, A. D. Rogol, W. S. Evans, L. A. Kolp, and M. L. Johnson. Operating characteristics of the male hypothalamo-pituitary-gonadal axis: pulsatile release of testosterone and follicle-stimulating hormone and their temporal coupling with luteinizing hormone. *J Clin Endocrinol Metab*, 65(5):929–41, 1987.

[155] M. Verhaegen and A. Hansson. Nuclear norm subspace identification (n2sid) for short data batches. *IFAC Proceedings Volumes*, 47(3):9528–9533, 2014.

[156] R. Vidal, S. Soatto, Y. Ma, and S. Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *42nd IEEE Conference on Decision and Control*, volume 1, pages 167–172. IEEE, 2003.

[157] J. Walker, J. Terry, K. Tsaneva-Atanasova, S. Armstrong, C. McArdle, and S. Lightman. Encoding and decoding mechanisms of pulsatile hormone secretion. *Journal of Neuroendocrinology*, 22:1226–1238, 2009.

[158] K. Werner, M. Jansson, and P. Stoica. On estimation of covariance matrices with Kronecker product structure. *IEEE Transactions on Signal Processing*, 56(2):478–491, 2008.

[159] N. Wiener. *Nonlinear problems in random theory*, volume 1. Wiley, New York, 1958.

[160] T. Wigren. *Recursive identification based on the nonlinear Wiener model*. PhD thesis, Uppsala University, Uppsala, Sweden, December 1990.

[161] T. Wigren. Recursive prediction error identification using the nonlinear Wiener model. *Automatica*, 29(4):1011 – 1025, 1993.

[162] T. Wigren. Convergence analysis of recursive identification algorithms based on the nonlinear Wiener model. *IEEE Transactions on Automatic Control*, 39(11):2191–2206, Nov 1994.

[163] T. Wigren. User choices and model validation in system identification using nonlinear Wiener models. In *13th IFAC Symposium on System Identification*, pages 863–868, 2003.

[164] T. Wigren. Recursive prediction error identification and scaling of non-linear state space models using a restricted black box parameterization. *Automatica*, 42(1):159 – 168, 2006.

[165] T. T. Wu and K. Lange. The MM alternative to EM. *Statistical Science*, 25(4):492–505, 2010.

[166] D. Zachariah and P. Stoica. Online hyperparameter-free sparse estimation method. *IEEE Transactions on Signal Processing*, 63(13):3348–3359, July 2015.

[167] W.-X. Zhao and H.-F. Chen. Recursive identification for Hammerstein system with ARX subsystem. *IEEE Transactions on Automatic Control*, 51(12):1966–1974, 2006.

[168] G. Zheng, Y. Orlov, W. Perruquetti, and J.-P. Richard. Finite time observer-based control of linear impulsive systems with persistently acting impact. *IFAC Proceedings Volumes*, 44(1):2442–2447, 2011.

[169] Z. T. Zhusubaliyev, A. N. Churilov, and A. Medvedev. Bifurcation phenomena in an impulsive model of non-basal testosterone regulation. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(1), 2012.

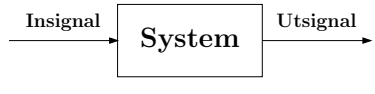
[170] M. Zitzmann and E. Nieschlag. Testosterone levels in healthy men and the relation to behavioural and physical characteristics: facts and constructs. *European Journal of Endocrinology*, 144(3):183–197, 2001.

Sammanfattning på svenska

Nedan följer en kort sammanfattning av avhandlingen, vars titel skulle kunna översättas till *"Modellering och identifiering av icke-linjära och impulsiva system"*.

Människor använder abstrakta modeller av verkliga objekt varje dag. Till exempel så har de flesta av oss en mental modell över hur en bil fungerar – "Om jag vrider ratten åt vänster, så åker bilen åt vänster. Om jag trycker på gasen, så åker bilen snabbare. Om jag trycker på bromsen, så åker bilen saktare. Och om jag tutar så säger bilen tut-tut". Denna modell av en bil säger inget om hur en bilmotor fungerar, men genom att bara beskriva hur förarens "insignaler" (rattläge, gas, broms och tuta) påverkar bilens "utsignaler" (position, fart och ljud) så har föraren nytta av den.

Om vi istället vill styra bilen med hjälp av en dator, så är det fördelaktigt att ha en matematisk modell som beskriver hur insignalerna påverkar utsignalerna. Problemet är att det som regel är väldigt svårt, för att inte säga omöjligt, att ta fram en sådan modell genom att använda fysikaliska och kemiska lagar för varje liten komponent i bilen. Ett sätt att lösa detta problem är att använda sig av systemidentifiering, vilket också är ämnet för första delen av denna avhandling.



Väldigt förenklat så innehåller systemidentifiering en uppsjö av metoder för att låta datorer analysera uppmätta in- och utsignaler, och sedan ta fram en matematisk modell för systemet som genererade dessa. Systemet kan här vara t.ex. en bil, en industrirobot, ett flygplan, eller människokroppen. Så, på detta sätt krävs det inte längre en djupgående analys av varje del i systemet, utan datorn kan, förhoppningsvis, lära sig en användbar modell från uppmätt data (in- och utsignaler). Vi säger att datorn identifierar en modell för systemet. En naiv lösning vore att

låta datorn ta fram en modell som beskriver uppmätt data exakt, men i praktiken innehåller uppmätt data alltid störningar av olika slag. Det kan bero på att våra mätinstrument inte är exakta, och att mer eller mindre slumpmässiga saker inträffade medan data samlades in (t.ex. kan vinden påverka bilens rörelse o.s.v.). Om vi vill att modellen ska vara användbar även för andra insignaler och störningar än de som användes under experimentet, så måste vår metod för att hitta en matematisk modell ta hänsyn till störningarna. Dessutom är begreppet matematiskt modell väldigt brett, och i praktiken måste vi begränsa oss till att leta efter en modell inom en väldefinierad klass av modeller. En populär sådan modellklass är de linjära modellerna. Lite förenklat har en linjär modell egenskapen att en dubblering av insignalen leder till en dubblering av utsignalen osv. Denna typ av modeller är relativt enkel att arbeta med och analysera, och därför finns det idag välbeprövade metoder inom systemidentifiering för att ta fram sådana modeller. Många system kan också beskrivas väl av en linjär modell, åtminstone så länge insignalerna håller sig inom vissa intervall. Men i praktiken är alla verkliga system icke-linjära – en dubblering av insignalen kommer inte alltid leda till en dubblering av utsignalen.

I den första delen av denna avhandling så tittar vi på hur olika typer av icke-linjära modeller kan hanteras. I Kapitel 2 ges en mer formell introduktion till icke-linjär systemidentifiering, och därefter följer tre kapitel där nya metoder för att identifiera icke-linjära modeller utvecklas.

I Kapitel 3 utvecklas ett nytt ramverk som kan användas för flera olika modellklasser. Här utgår vi från en nominell modell, som kan kan antas vara linjär. När data sedan mäts upp så kommer den föreslagna metoden automatiskt förfina den nominella modellen och, om uppmätta data tyder på att det behövs, lägga in icke-linjära element. För att den förfinade modellen inte ska bli allt för komplicerad så är metoden utvecklad så att enklare modeller kommer att föredras så länge de förklarar uppmätt data tillräckligt bra. Eftersom störningar ofta påverkar systemet på ett slumpmässigt sätt, så bygger hela ramverket på stokastiska modeller, dvs. modeller som baseras på sannolikheter. Givet en sådan modell och en insignal, går det att räkna ut vad sannolikheten för en viss utsignal är enligt modellen. Den föreslagna metoden försöker därför hitta den modell som ger störst sannolikhet för att den uppmätta utsignalen skulle observeras, detta kallas “maximum likelihood”-metoden. Av alla modeller vill vi alltså hitta den modell som ger störst sannolikhet för att vi skulle ha observerat det vi faktiskt observerade.

I Kapitel 4 visas sedan att bitvis linjära modeller kan identifieras med hjälp av ramverket som utvecklas i Kapitel 3. Bitvis linjära modeller baseras på linjära modeller, men här tillåter vi att olika linjära modeller användes beroende på t.ex. hur stor insignalen är. Bitvis linjära

modeller är väldigt generella, och kan approximera andra icke-linjära modeller godtyckligt väl. De flesta verkliga system har egenskapen att effekten av ökad/minskad insignal avtar för stora/små insignaler. Till exempel, när du trycker hårdare på gaspedalen i bilen så kommer farten öka, men det finns en gräns där hårdare tryck på gaspedalen inte längre leder till ökad fart (gasen i botten!). En bitvis linjär modell ta hänsyn till sådana effekter, till skillnad från en vanlig linjär modell där ökad insignal alltid leder till ökad utsignal.

Ett annat sätt att ta hänsyn till att olika storlekar på insignalen påverkar systemet på olika sätt, är att först skicka insignalen genom en icke-linjär del, och sedan vidare till en linjär modell. En modell uppbyggd på detta sätt kallas för en Hammerstein-modell. I Kapitel 5 visas hur den icke-linjära delen och den linjära delen kan identifieras samtidigt. Där görs även en utförlig analys som visar att metoden, givet vissa antaganden, kommer att hitta den modell som (lokalt) minimerar prediktionsfelet ifall tillräckligt mycket data samlas in.

I första delen av avhandlingen är metoderna inte utvecklade med något specifikt system i åtanke, men samtliga metoder har testats på verkliga system. I andra delen av avhandlingen studeras modellering av testosteron-reglering hos män.

Testosteron spelar hos män en central roll i det reproduktiva systemet, men är också relaterat till muskelmassa, fettvävnad med mera. Vid reglering av testosteron (Te) så spelar även två andra hormoner en central roll, nämligen gonadotropinfrisättande hormon (GnRH) och luteiniserande hormon (LH). GnRH utsöndras från hypotalamus i hjärnan, och stimulerar sedan utsöndring av LH i hypofysen. LH tar sig sedan via blodet till testiklarna där det stimulerar utsöndring av Te. Loopen sluts genom att Te i sin tur undertrycker utsöndringen av GnRH, se Figur 6.1 i Kapitel 6 för en schematisk illustration av detta.

Testosteron-regleringen är en del av kroppens endokrina system, alltså de delar av kroppen som ansvarar för hormonreglering. Att Te-reglering kan beskrivas på ett relativt enkelt sätt medför att detta är en god kandidat för ett första försök till modellering av det endokrina systemet, och förhoppningen är att metoder som utvecklas i andra delen av avhandlingen i framtiden kan användas för andra delar av det endokrina systemet.

I Kapitel 6 ges en introduktion till modellering av testosteron, och en impulsiv modell som tidigare använts beskrivs. Modellen är impulsiv för att kunna beskriva utsöndringen av GnRH, som sker i impulser då en stor mängd GnRH utsöndras under väldigt kort tid. En nackdel men den tidigare modellen är att den inte tar hänsyn till att det tar en viss tid för LH att färdas från hjärnan ner till testiklarna. Därför utökar vi denna modell i Kapitel 7 och Kapitel 9 genom att bland annat inkludera en tidsfördröjning. Med en tidsfördröjning i systemet blir det som regel svårare att analysera vilka typer av beteenden som kan uppstå

i modellen, men i Kapitel 7 visas hur tidsfördjningen kan hanteras på ett sätt som gör att metoder för fallet utan tidsfördröjning fortfarande kan användas.

I modellen ingår ett antal parametrar som kan vara olika från person till person, så det är av intresse att kunna skatta dessa parametrar för en person genom att mäta hur koncentrationerna av de olika hormonerna varierar över tid. Ett problem här är att det inte går att mäta upp koncentrationen av GnRH i kliniska studier utan att göra ingrepp i hjärnan, och därför går det heller inte mäta exakt när impulserna inträffar. I Kapitel 8 och Kapitel 9 utvecklas därför metoder för att skatta GnRH-impulserna från uppmätta värden av LH-koncentrationen. I Kapitel 9 används sedan metoder från systemidentifiering för att skatta alla parametrar med hjälp av uppmäta koncentrationer av LH och Te. Där jämförs också det beteende som modellen uppvisar med de kliniskt uppmätta hormonkoncentrationerna.

I Kapitel 10 visas sedan hur effekten av externa signaler kan analyseras med hjälp av modellen. Dessa externa signaler kan representera andra hormoner som påverkar Te-regleringen, effekter som följer på grund av dygnsrytmen, eller hur olika läkemedel påverkar kroppens Te-reglering.

Acta Universitatis Upsaliensis
Uppsala Dissertations from the Faculty of Science
Editor: The Dean of the Faculty of Science

1–11: 1970–1975

12. *Lars Thofelt*: Studies on leaf temperature recorded by direct measurement and by thermography. 1975.
13. *Monica Henricsson*: Nutritional studies on Chara globularis Thuill., Chara zeylanica Willd., and Chara haitensis Turpin. 1976.
14. *Göran Kloow*: Studies on Regenerated Cellulose by the Fluorescence Depolarization Technique. 1976.
15. *Carl-Magnus Backman*: A High Pressure Study of the Photolytic Decomposition of Azoethane and Propionyl Peroxide. 1976.
16. *Lennart Källströmer*: The significance of biotin and certain monosaccharides for the growth of Aspergillus niger on rhamnose medium at elevated temperature. 1977.
17. *Staffan Renlund*: Identification of Oxytocin and Vasopressin in the Bovine Adenohypophysis. 1978.
18. *Bengt Finnström*: Effects of pH, Ionic Strength and Light Intensity on the Flash Photolysis of L-tryptophan. 1978.
19. *Thomas C. Amu*: Diffusion in Dilute Solutions: An Experimental Study with Special Reference to the Effect of Size and Shape of Solute and Solvent Molecules. 1978.
20. *Lars Tegnér*: A Flash Photolysis Study of the Thermal Cis-Trans Isomerization of Some Aromatic Schiff Bases in Solution. 1979.
21. *Stig Tormod*: A High-Speed Stopped Flow Laser Light Scattering Apparatus and its Application in a Study of Conformational Changes in Bovine Serum Albumin. 1985.
22. *Björn Varnestig*: Coulomb Excitation of Rotational Nuclei. 1987.
23. *Frans Lettenström*: A study of nuclear effects in deep inelastic muon scattering. 1988.
24. *Göran Ericsson*: Production of Heavy Hypernuclei in Antiproton Annihilation. Study of their decay in the fission channel. 1988.
25. *Fang Peng*: The Geopotential: Modelling Techniques and Physical Implications with Case Studies in the South and East China Sea and Fennoscandia. 1989.
26. *Md. Anowar Hossain*: Seismic Refraction Studies in the Baltic Shield along the Fennolora Profile. 1989.
27. *Lars Erik Svensson*: Coulomb Excitation of Vibrational Nuclei. 1989.
28. *Bengt Carlsson*: Digital differentiating filters and model based fault detection. 1989.
29. *Alexander Edgar Kavka*: Coulomb Excitation. Analytical Methods and Experimental Results on even Selenium Nuclei. 1989.
30. *Christopher Juhlin*: Seismic Attenuation, Shear Wave Anisotropy and Some Aspects of Fracturing in the Crystalline Rock of the Siljan Ring Area, Central Sweden. 1990.

31. *Torbjörn Wigren*: Recursive Identification Based on the Nonlinear Wiener Model. 1990.
32. *Kjell Janson*: Experimental investigations of the proton and deuteron structure functions. 1991.
33. *Suzanne W. Harris*: Positive Muons in Crystalline and Amorphous Solids. 1991.
34. *Jan Blomgren*: Experimental Studies of Giant Resonances in Medium-Weight Spherical Nuclei. 1991.
35. *Jonas Lindgren*: Waveform Inversion of Seismic Reflection Data through Local Optimisation Methods. 1992.
36. *Liqi Fang*: Dynamic Light Scattering from Polymer Gels and Semidilute Solutions. 1992.
37. *Raymond Munier*: Segmentation, Fragmentation and Jostling of the Baltic Shield with Time. 1993.

Prior to January 1994, the series was called *Uppsala Dissertations from the Faculty of Science*.

Acta Universitatis Upsaliensis

Uppsala Dissertations from the Faculty of Science and Technology
Editor: The Dean of the Faculty of Science

1–14: 1994–1997. 15–21: 1998–1999. 22–35: 2000–2001. 36–51: 2002–2003.

52. *Erik Larsson*: Identification of Stochastic Continuous-time Systems. Algorithms, Irregular Sampling and Cramér-Rao Bounds. 2004.
53. *Per Åhgren*: On System Identification and Acoustic Echo Cancellation. 2004.
54. *Felix Wehrmann*: On Modelling Nonlinear Variation in Discrete Appearances of Objects. 2004.
55. *Peter S. Hammerstein*: Stochastic Resonance and Noise-Assisted Signal Transfer. On Coupling-Effects of Stochastic Resonators and Spectral Optimization of Fluctuations in Random Network Switches. 2004.
56. *Esteban Damián Avendaño Soto*: Electrochromism in Nickel-based Oxides. Coloration Mechanisms and Optimization of Sputter-deposited Thin Films. 2004.
57. *Jenny Öhman Persson*: The Obvious & The Essential. Interpreting Software Development & Organizational Change. 2004.
58. *Chariklia Rouki*: Experimental Studies of the Synthesis and the Survival Probability of Transactinides. 2004.
59. *Emad Abd-Elrady*: Nonlinear Approaches to Periodic Signal Modeling. 2005.
60. *Marcus Nilsson*: Regular Model Checking. 2005.
61. *Pritha Mahata*: Model Checking Parameterized Timed Systems. 2005.
62. *Anders Berglund*: Learning computer systems in a distributed project course: The what, why, how and where. 2005.
63. *Barbara Piechocinska*: Physics from Wholeness. Dynamical Totality as a Conceptual Foundation for Physical Theories. 2005.
64. *Pär Samuelsson*: Control of Nitrogen Removal in Activated Sludge Processes. 2005.

65. *Mats Ekman*: Modeling and Control of Bilinear Systems. Application to the Activated Sludge Process. 2005.
66. *Milena Ivanova*: Scalable Scientific Stream Query Processing. 2005.
67. *Zoran Radovic´*: Software Techniques for Distributed Shared Memory. 2005.
68. *Richard Abrahamsson*: Estimation Problems in Array Signal Processing, System Identification, and Radar Imagery. 2006.
69. *Fredrik Robelius*: Giant Oil Fields – The Highway to Oil. Giant Oil Fields and their Importance for Future Oil Production. 2007.
70. *Anna Davour*: Search for low mass WIMPs with the AMANDA neutrino telescope. 2007.
71. *Magnus Ågren*: Set Constraints for Local Search. 2007.
72. *Ahmed Rezine*: Parameterized Systems: Generalizing and Simplifying Automatic Verification. 2008.
73. *Linda Brus*: Nonlinear Identification and Control with Solar Energy Applications. 2008.
74. *Peter Nauclér*: Estimation and Control of Resonant Systems with Stochastic Disturbances. 2008.
75. *Johan Petrini*: Querying RDF Schema Views of Relational Databases. 2008.
76. *Noomene Ben Henda*: Infinite-state Stochastic and Parameterized Systems. 2008.
77. *Samson Keleta*: Double Pion Production in dd→αππ Reaction. 2008.
78. *Mei Hong*: Analysis of Some Methods for Identifying Dynamic Errors-invariables Systems. 2008.
79. *Robin Strand*: Distance Functions and Image Processing on Point-Lattices With Focus on the 3D Face-and Body-centered Cubic Grids. 2008.
80. *Ruslan Fomkin*: Optimization and Execution of Complex Scientific Queries. 2009.
81. *John Airey*: Science, Language and Literacy. Case Studies of Learning in Swedish University Physics. 2009.
82. *Arvid Pohl*: Search for Subrelativistic Particles with the AMANDA Neutrino Telescope. 2009.
83. *Anna Danielsson*: Doing Physics – Doing Gender. An Exploration of Physics Students' Identity Constitution in the Context of Laboratory Work. 2009.
84. *Karin Schönning*: Meson Production in pd Collisions. 2009.
85. *Henrik Petrén*: η Meson Production in Proton-Proton Collisions at Excess Energies of 40 and 72 MeV. 2009.
86. *Jan Henry Nyström*: Analysing Fault Tolerance for ERLANG Applications. 2009.
87. *John Håkansson*: Design and Verification of Component Based Real-Time Systems. 2009.
88. *Sophie Grape*: Studies of PWO Crystals and Simulations of the $\bar{p}p \to \bar{\Lambda}\Lambda, \bar{\Lambda}\Sigma^0$ Reactions for the PANDA Experiment. 2009.
90. *Agnes Rensfelt*. Viscoelastic Materials. Identification and Experiment Design. 2010.
91. *Erik Gudmundson*. Signal Processing for Spectroscopic Applications. 2010.
92. *Björn Halvarsson*. Interaction Analysis in Multivariable Control Systems. Applications to Bioreactors for Nitrogen Removal. 2010.
93. *Jesper Bengtson*. Formalising process calculi. 2010.
94. *Magnus Johansson*. Psi-calculi: a Framework for Mobile Process Calculi. Cook your own correct process calculus – just add data and logic. 2010.
95. *Karin Rathsman*. Modeling of Electron Cooling. Theory, Data and Applications. 2010.

96. *Liselott Dominicus van den Bussche*. Getting the Picture of University Physics. 2010.
97. *Olle Engdegård*. A Search for Dark Matter in the Sun with AMANDA and IceCube. 2011.
98. *Matthias Hudl.* Magnetic materials with tunable thermal, electrical, and dynamic properties. An experimental study of magnetocaloric, multiferroic, and spin-glass materials. 2012.
99. *Marcio Costa.* First-principles Studies of Local Structure Effects in Magnetic Materials. 2012.
100. *Patrik Adlarson*. Studies of the Decay $\eta \rightarrow \pi^+\pi^-\pi^0$ with WASA-at-COSY. 2012.
101. *Erik Thomé.* Multi-Strange and Charmed Antihyperon-Hyperon Physics for PANDA. 2012.
102. *Anette Löfström.* Implementing a Vision. Studying Leaders' Strategic Use of an Intranet while Exploring Ethnography within HCI. 2014.
103. *Martin Stigge.* Real-Time Workload Models: Expressiveness vs. Analysis Efficiency. 2014.
104. *Linda Åmand.* Ammonium Feedback Control in Wastewater Treatment Plants. 2014.
105. *Mikael Laaksoharju.* Designing for Autonomy. 2014.
106. *Soma Tayamon.* Nonlinear System Identification and Control Applied to Selective Catalytic Reduction Systems. 2014.