EMAD ABD-ELRADY

Nonlinear Approaches to Periodic Signal Modeling

UPPSALA UNIVERSITET

ABSTRACT


Periodic signal modeling plays an important role in different fields since many physical signals can be considered to be approximately periodic. Examples include speech, musical waveforms, acoustic waves, signals recorded during patient monitoring and outputs of possibly nonlinear systems excited by a sinusoidal input.

The unifying theme of this thesis is using nonlinear techniques to model periodic signals. The suggested techniques utilize the user pre-knowledge about the signal waveform. This gives the suggested techniques an advantage, provided that the assumed model structure is suitable, as compared to other approaches that do not consider such priors.

The suggested technique of the first part of this thesis relies on the fact that a sine wave that is passed through a static nonlinear function produces a harmonic spectrum of overtones. Consequently, the estimated signal model can be parameterized as a known periodic function (with unknown frequency) in cascade with an unknown static nonlinearity. The unknown frequency and the parameters of the static nonlinearity (chosen as the nonlinear function values in a set of fixed grid points) are estimated simultaneously using the recursive prediction error method (RPEM).

A complete treatment of the local convergence properties of the suggested RPEM algorithm is provided. Also, an adaptive grid point algorithm is introduced to estimate the unknown frequency and the parameters of the static nonlinearity in a number of adaptively estimated grid points. This gives the RPEM method more freedom to select the grid points and hence reduces modeling errors.

Limit cycle oscillations problem are encountered in many applications. Therefore, mathematical modeling of limit cycles becomes an essential topic that helps to better understand and/or to avoid limit cycle oscillations in different fields.

In the second part of this thesis, a second-order nonlinear ODE model is used to model the periodic signal as a limit cycle oscillation. The right hand side of the suggested ODE model is parameterized using a polynomial function in the states, and then discretized to allow for the implementation of different identification algorithms. Hence, it is possible to obtain highly accurate models by only estimating a few parameters. Also, this is conditioned on the fact that the signal model is suitable.

In the third part, different user aspects for the two nonlinear approaches of the thesis are discussed. Also, some comments on other approaches to the problem of modeling periodic signals are given. Finally, topics for future research are presented.

Keywords: Cramér-Rao bounds; frequency estimation; identification; limit cycle; nonlinear systems, periodic signals; Wiener model structure.

Emad Abd-Elrady, Uppsala University, Department of Information Technology, Division of Systems and Control, P. O. Box 337, SE-751 05 Uppsala, Sweden.

© Emad Abd-Elrady 2005

ISSN 1104-2516
urn:nbn:se:uu:diva-4644 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-4644)
Printed in Sweden by Elanders Gotab, Stockholm 2005
Distributor: Uppsala University Library, Box 510, SE-751 20 Uppsala, Sweden
To my family
Acknowledgements

First of all I would like to express my deep and sincere gratitude to my supervisors Prof. Torsten Söderström and Prof. Torbjörn Wigren for their unlimited support, excellent guidance and for sharing their profound knowledge in system identification. I am also grateful to them for valuable comments and suggestions on the manuscript of this thesis.

I am indebted to Prof. Johan Schoukens for valuable discussions and pleasant collaboration.

I am thankful to Ms. Ingegerd Wås for her patience and understanding while helping with all administrative issues.

I wish also to thank all my colleagues at Systems and Control group for creating such a pleasant work environment, stimulating discussions and for their great tendency to help. I am indebted to Mats Ekman for interesting discussions and for his useful comments on the manuscript of this thesis. Also, I am thankful to Peter Nauclér and Agnes Runqvist for providing the Swedish translation of the thesis summary.

I am thankful to Prof. Jonas Sjöberg for serving as my faculty opponent. Thanks are due to Prof. Irene Gu, Prof. Xiaoming Hu and Assoc. Prof. Peter Händel for being the members of my thesis committee.

I am grateful to Prof. Atef Bassiouney and Prof. Eweda Eweda for their support and encouragement during my master study in Egypt. I would like also to thank my colleagues at the Higher Technological Institute for their help and friendship.

Also, my friends in Uppsala deserve a special thank for their support and friendship. I am indebted to Usama Hegazy, Mohamed Issa, Dr. Idress Attitalla, and their families for the nice time that we spent together.

My parents, my sisters and their families, and my wife’s family have been always encouraging me during my study. My wife Riham has been always understanding and supportive. This thesis would not exist without her love and patience. Finally, I am grateful to my kids, Hadil and Omar, for adding a big smile to my life.
# Contents

<table>
<thead>
<tr>
<th>Glossary</th>
<th>xiii</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 Introduction</strong></td>
<td></td>
</tr>
<tr>
<td>1.1 Periodic Signals</td>
<td></td>
</tr>
<tr>
<td>1.1.1 Fourier Series Representation</td>
<td></td>
</tr>
<tr>
<td>1.1.2 Importance of Periodic Signal Modeling</td>
<td></td>
</tr>
<tr>
<td>1.1.3 Introductory Examples</td>
<td></td>
</tr>
<tr>
<td>1.2 Modeling of Periodic Signals</td>
<td></td>
</tr>
<tr>
<td>1.2.1 System Identification</td>
<td></td>
</tr>
<tr>
<td>1.2.2 The Scope of the Thesis</td>
<td></td>
</tr>
<tr>
<td>1.3 Previous Approaches to Periodic Signal Modeling</td>
<td></td>
</tr>
<tr>
<td>1.3.1 The Periodogram</td>
<td></td>
</tr>
<tr>
<td>1.3.2 Modeling of Line-Spectra</td>
<td></td>
</tr>
<tr>
<td>1.3.3 The Adaptive Comb Filter</td>
<td></td>
</tr>
<tr>
<td>1.4 Periodic Signal Modeling Based on the Wiener Model Structure</td>
<td></td>
</tr>
<tr>
<td>1.4.1 Block-Structured Models</td>
<td></td>
</tr>
<tr>
<td>1.4.2 The Adaptive Nonlinear Modeler</td>
<td></td>
</tr>
<tr>
<td>1.4.3 The Periodic Signal Modeling Approach Based on the Wiener Model</td>
<td></td>
</tr>
<tr>
<td>1.5 Periodic Signal Modeling Using Orbits of Second-Order Nonlinear ODE’s</td>
<td></td>
</tr>
<tr>
<td>1.5.1 Dynamical Systems</td>
<td></td>
</tr>
<tr>
<td>1.5.2 Nonlinear Models of Dynamical Systems</td>
<td></td>
</tr>
<tr>
<td>1.5.3 Solutions of Ordinary Differential Equations</td>
<td></td>
</tr>
<tr>
<td>1.5.4 Second-Order Autonomous Systems</td>
<td></td>
</tr>
<tr>
<td>1.5.5 Limit Cycles</td>
<td></td>
</tr>
<tr>
<td>1.5.6 Stability of Periodic Solutions</td>
<td></td>
</tr>
<tr>
<td>1.5.7 The Periodic Signal Modeling Approach Using Periodic Orbits</td>
<td></td>
</tr>
<tr>
<td>1.6 Why and When to Use the Proposed Methods in this Thesis?</td>
<td></td>
</tr>
<tr>
<td>1.7 Thesis Outline and Contributions</td>
<td></td>
</tr>
</tbody>
</table>
## I Periodic Signal Modeling Based on the Wiener Model Structure

### 2 Periodic Signal Modeling Using Adaptive Nonlinear Function Estimation

#### 2.1 Introduction

#### 2.2 Review of the Algorithm of [121]

#### 2.3 Piecewise Quadratic Approximation

#### 2.4 Numerical Examples

#### 2.5 Conclusions

### 3 A Modified Nonlinear Approach to Periodic Signal Modeling

#### 3.1 Introduction

#### 3.2 The Modified Algorithm

#### 3.3 Local Convergence

##### 3.3.1 The Associated ODE Approach

##### 3.3.2 Analysis Details

##### 3.3.3 Discussion

#### 3.4 The Cramér-Rao Bound

#### 3.5 Numerical Examples

#### 3.6 Conclusions

#### 3.A Proof of Lemma 3.2

#### 3.B Proof of Lemma 3.3

#### 3.C Proof of Theorem 3.2

### 4 An Adaptive Grid Point Algorithm for Periodic Signal Modeling

#### 4.1 Introduction

#### 4.2 The Suggested Algorithm

#### 4.3 The Cramér-Rao Bound

#### 4.4 Numerical Examples

#### 4.5 Conclusions

#### 4.A Proof of Theorem 4.1

## II Periodic Signal Modeling Using Orbits of Second-Order Nonlinear ODE’s

### 5 The Second-Order Nonlinear ODE Model

#### 5.1 Introduction

#### 5.2 Examples
### CONTENTS

5.3 Modeled Signal and Measurements ........................................ 107
5.4 Model Structures ......................................................... 107
5.5 Parameterization ......................................................... 109
5.6 Discretization ............................................................. 110

6 Sufficiency of Second-Order Nonlinear ODE for Modeling Periodic Signals ......................................................... 111
6.1 Introduction ............................................................... 111
6.2 General Assumptions on the Modeled Signal ......................... 112
6.3 Smoothness of the Modeled Orbit ..................................... 112
6.4 Conditions on $y(t)$ to Be a Solution ................................. 113
6.5 Uniqueness of the Solution ............................................... 116
6.6 When a Second-Order ODE Model Is Insufficient .................. 117
6.7 Conclusions ............................................................... 118

7 Least Squares and Markov Estimation of Periodic Signals ............. 119
7.1 Introduction ............................................................... 119
7.2 The Least Squares Algorithm ............................................ 120
7.3 The Markov Estimate ...................................................... 122
7.4 Numerical Examples ...................................................... 124
7.4.1 The LS Algorithm Simulation Study ............................... 125
7.4.2 The Markov Estimate Compared to the LS Estimate ............ 127
7.5 Conclusions ............................................................... 130

8 Periodic Signal Modeling with Kalman Filters .......................... 131
8.1 Introduction ............................................................... 131
8.2 The Kalman Filter (KF) .................................................. 132
8.3 The Extended Kalman Filter (EKF) .................................... 134
8.4 Simulation Study ......................................................... 134
8.5 Conclusions ............................................................... 138

9 Maximum Likelihood Estimation of Periodic Signals .................... 139
9.1 Introduction ............................................................... 139
9.2 The ODE models ......................................................... 139
9.3 The Maximum Likelihood Method ..................................... 141
9.4 The Cramér-Rao Bound ................................................ 143
9.5 Simulation Study ......................................................... 144
9.6 Conclusions ............................................................... 149
9.A Derivation of the Details of the CRB ................................. 149
## User Aspects and Future Research

### 13 User Aspects

13.1 Introduction

13.2 User Aspects for the Approach of Part I

13.2.1 Modeled Signal

13.2.2 Choice of the Algorithm

13.2.3 Grid Points

13.2.4 Nonlinear Modeler

13.2.5 Additive Noise

13.2.6 Local Minima

13.2.7 Design Variables

13.2.8 Computational Complexity

13.3 User Aspects for the Approach of Part II

13.3.1 Modeled Signal

13.3.2 Choice of the Algorithm

13.3.3 Sampling Period

13.3.4 Additive Noise

13.3.5 Design Variables

13.3.6 Polynomial Degrees

13.3.7 Computational Complexity

### 14 Conclusions and Future Research

14.1 Conclusions

14.2 Topics for Future Research

---

Svensk Sammanfattning

Bibliography
Glossary

The following lists of notations and abbreviations are intended to introduce symbols that are frequently used in the thesis. The notational convention used is that matrices are written with bold face, upper case, for example, \( \mathbf{R} \) and that vectors are written with bold face, lower case, for example, \( \mathbf{r} \). Note that the same symbol may occasionally be used for a different purpose.

**Notations**

- **\( \text{col } \mathbf{R} \)**: a column vector created by stacking the columns of the matrix \( \mathbf{R} \) on top of each other
- **\( D \)**: differentiation operator
- **\( D_M \)**: set of parameter vectors describing the model set
- **\( E \)**: expectation operator
- **\( e(kh) \)**: discrete-time white noise
- **\( e(t) \)**: continuous-time white noise
- **\( f \)**: frequency
- **\( f_s \)**: sampling frequency
- **\( h \)**: sampling interval
- **\( I \)**: identity matrix
- **\( \inf \)**: the infimum, the greatest lower bound
- **\( \max_\theta \)**: maximization with respect to \( \theta \)
- **\( \min_\theta \)**: minimization with respect to \( \theta \)
- **\( N \)**: number of measurements
- **\( \mathbb{N} \)**: set of positive integers
- **\( \mathcal{N}(0, \sigma^2) \)**: normal distribution with zero mean and variance \( \sigma^2 \)
- **\( \mathcal{O}(h) \)**: \( \Delta = \mathcal{O}(h) \Rightarrow \Delta/h \) is bounded when \( h \to 0 \)
- **\( p(x) \)**: probability density function of \( x \)
- **\( q \)**: forward-shift operator \((qy(kh) = y(kh + h))\)
- **\( q^{-1} \)**: backward-shift operator \((q^{-1}y(kh) = y(kh - h))\)
- **\( R \)**: set of real numbers
- **\( \mathbb{R}^n \)**: the \( n \)-dimensional Euclidean space
- **\( \mathbf{R}_{i \times j} \)**: a matrix of \( i \) rows and \( j \) columns
- **\( \mathbf{R}^T \)**: transpose of the matrix \( \mathbf{R} \)
- **\( \mathbf{R}^\dagger \)**: pseudoinverse of the matrix \( \mathbf{R} \)
- **\( \sup \)**: the supremum, the least upper bound
- **\( T \)**: time period
- **\( \dot{x} \)**: time derivative of \( x(t) \)
- **\( \hat{x} \)**: estimate of \( x \)
- **\( |x| \)**: absolute value of \( x \)
∥\mathbf{x}\parallel \quad \text{the } L_2 \text{ norm of the vector } \mathbf{x} (\|\mathbf{x}\|^2 = \mathbf{x}^T \mathbf{x})

\mathbb{Z} \quad \text{set of all integers}

\mathbb{Z}^N \quad \text{set of measurements}

\delta_{k,s} \quad \text{Kronecker delta (}= 1 \text{ if } k = s, \text{ otherwise } = 0)

\delta(\tau) \quad \text{Dirac’s } \delta\text{-function}

\omega \quad \text{angular frequency (}\omega = \frac{2\pi}{T})

\emptyset \quad \text{empty set}

\in \quad \text{belongs to}

\subset \quad \text{subset}

\cap \quad \text{intersection}

\cup \quad \text{union}

\triangleq \quad \text{equal by definition}

\neq \quad \text{not equal}

\approx \quad \text{approximately equal}

\Rightarrow \quad \text{implies}

\exists \quad \text{exists}

\forall \quad \text{for all}

Abbreviations

AR \quad \text{autoregressive}

ARMA \quad \text{autoregressive moving average}

ASA \quad \text{automated spectral analysis}

BLUE \quad \text{best linear unbiased estimate}

cf. \quad \text{confere, compare}

CRB \quad \text{Cramér-Rao bound}

dB \quad \text{decibel}

DFT \quad \text{discrete Fourier transform}

EB \quad \text{Euler backward approximation}

EC \quad \text{Euler center approximation}

EF \quad \text{Euler forward approximation}

e.g. \quad \text{exempli gratia, for the sake of example}

EKF \quad \text{extended Kalman filter}

EIV \quad \text{error in variables}

ESPRIT \quad \text{estimation of signal parameters by rotational invariance techniques}

FFT \quad \text{fast Fourier transform}

FIR \quad \text{finite impulse response}

i.e. \quad \text{id est, that is}

IIR \quad \text{infinite impulse response}

KF \quad \text{Kalman filter}

LHP \quad \text{left half plane}

LS \quad \text{least squares}

MISO \quad \text{multiple input, single output}
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML</td>
<td>maximum likelihood</td>
</tr>
<tr>
<td>MSE</td>
<td>mean squared error</td>
</tr>
<tr>
<td>MUSIC</td>
<td>multiple signal classification</td>
</tr>
<tr>
<td>ODE</td>
<td>ordinary differential equation</td>
</tr>
<tr>
<td>pdf</td>
<td>probability density function</td>
</tr>
<tr>
<td>PSD</td>
<td>power spectral density</td>
</tr>
<tr>
<td>RHP</td>
<td>right half plane</td>
</tr>
<tr>
<td>RLS</td>
<td>recursive least squares</td>
</tr>
<tr>
<td>RPEACF</td>
<td>recursive prediction error adaptive comb filter</td>
</tr>
<tr>
<td>RPEM</td>
<td>recursive prediction error method</td>
</tr>
<tr>
<td>SISO</td>
<td>single input, single output</td>
</tr>
<tr>
<td>SNR</td>
<td>signal to noise ratio</td>
</tr>
<tr>
<td>TLS</td>
<td>total least squares</td>
</tr>
<tr>
<td>WELS</td>
<td>weighted extended least squares</td>
</tr>
<tr>
<td>w.r.t.</td>
<td>with respect to</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Periodic Signals

NATURE is full of systems which return periodically to the same initial state, passing through the same sequence of intermediate states every period. Everyday life is more or less built up around such phenomena, from night and day to the rise and fall of the tides, the phases of the moon, and the annual cycle of the seasons.

A signal \( f_p(t) \) is said to be periodic if there is a positive real number \( T_o \) such that for all times \( t \),

\[
f_p(t) = f_p(t + T_o).
\]

The smallest value of \( T_o \) that achieves (1.1) is called the period of \( f_p(t) \). Some examples of periodic signals are shown in Fig. 1.1.

Many physical signals can be considered to be approximately periodic. Examples include speech, musical waveforms, acoustic waves generated by helicopters and boats, and outputs of possibly nonlinear systems excited by a sinusoidal input, see [75, 121].

In biomedicine, several signals recorded during patient monitoring are approximately periodic, e.g. heart rhythm, cardiograms and pneumograms, see [1]. In repetitive control applications, e.g. doing some repetitive jobs using robotic manipulators, periodic signals can be used as a reference control signal, see [64, 100]. In mechanical engineering, periodic signals can be used in the detection and estimation of machine vibration, see [124]. In signal processing, the problem of extracting signals that contain (almost) periodic components from the noise corrupted observations arises in radar, sonar, and communications where it is important to estimate the parameters of these periodic components, see [102].

1.1.1 Fourier Series Representation

There is one sort of periodic behaviour that can be regarded as the basis for describing more general cases. This is the “sinusoidal” waveform shown in Fig. 1.1(a), so called because one realization is the sine function, \( \sin(t) \).

General periodic signals can be represented by a sum of sinusoidal waves whose frequencies are integral multiples of the lowest frequency (the so-called
1. Introduction

fundamental frequency). This representation is known as a Fourier series, see e.g. [81], and it is given by

\[ f_p(t) = C_0 + \sum_{k=1}^{\infty} C_k \sin(k\omega_o t + \phi_k), \]  

where \( C_0 \) is the mean value of the periodic signal, \( \omega_o = 2\pi/T_o \) is the fundamental angular frequency, \( C_k \) and \( \phi_k \) are the amplitude and phase of the \( k \)th harmonic component of \( f_p(t) \), respectively. For example, a square wave, see Fig. 1.1(b), can be approximated by

\[ s(t) = \sum_{k=1,3,5,...}^{\infty} \frac{1}{k} \sin(k\omega_o t). \]  

The approximated square wave is shown in Fig. 1.2 using 3 and 8 harmonics.
1.1. Periodic Signals

1.1.2 Importance of Periodic Signal Modeling

Periodic signal modeling plays an important role in many fields, e.g., for the following reasons:

• Tracking of time-varying parameters of sinusoids in additive noise is of great importance in many engineering applications such as radar, communications, control, biomedical engineering and others, see e.g. [34, 42, 43, 68, 106, 115].

• In many applications of signal processing as in sonar, radar and communications, it is desirable to eliminate or extract sine waves from observed data or to estimate their unknown frequencies, amplitudes and phases, see e.g. [33, 46, 48, 49, 102].

• In numerous practical applications there is a need to enhance or eliminate signals locally modeled as a finite or infinite sum of harmonics. These include enhancement of noisy biological signals, such as voiced speech and heart waveforms. When the parameters of the signal are unknown or possibly time-varying a natural approach is to use an adaptive filter to enhance/eliminate the signal or to estimate its parameters, see e.g. [20, 51, 75].

• The design of power systems is complicated in the presence of harmonics. Presence of harmonics can lead to unpredicted interaction between components in power networks, which in the worst case can lead to instability, see e.g. [3, 73].

• Periodic signal models have been found to be very good candidates for use in speech synthesis systems or in speech production machines. Text-to-speech (TTS) systems are used for voice delivery of text messages and
e-mail, reading books and voice response to database inquiries, see e.g. [32, 79].

1.1.3 Introductory Examples

Example 1.1. Number of hours of daylight.

Many important periodic phenomena abound in the real world. The number of hours of daylight on a given day of the year at any particular location on earth is a periodic function of time. It can be modeled by (see [36])

\[ N(t) = n_o + \sigma \sin \left( \frac{2\pi(t - t_{n_o})}{T_o} \right), \]  

(1.4)

where

• \( t \) is the number of days from January 1 of any given year,
• \( n_o \) is the average number of hours of daylight (12 hours),
• \( \sigma \) is the maximum variation above (in summer) and below (in winter) \( n_o \). This means that the longest day is \( n_o + \sigma \) hours and the shortest day is \( n_o - \sigma \) hours,
• \( t_{n_o} \) is the number of the day when exactly \( n_o \) hours of daylight occur,
• \( T_o \) is the period of the periodic function \( N(t) \) which in this case equals the number of days in a year (365 days).

Once the function (1.4) is available, it is possible to answer a variety of questions such as: How many hours of daylight would be expected on a particular date? When will there be a specific number of hours of daylight? For example, the plot of the function \( N(t) \) is given in Fig. 1.3 for Stockholm.
Example 1.2. The pumping heart.

The human heart, a pump which takes re-oxygenated blood from the lungs and pumps it out to the rest of the body, may be modeled as an oscillator. The system oscillates between two states: diastole, or relaxed state, and systole, or contracted state. An electro-chemical stimulus causes the heart muscle contraction and transition from diastole to systole states. A simplified model of this process is the modified Van der Pol oscillator, see [14]:

\[
\begin{align*}
\dot{x} &= \mu \left[ -v - \left( \frac{x^3}{3} - \alpha x \right) \right], \\
\dot{v} &= \frac{x - x_0}{\mu},
\end{align*}
\]

where \(x(t)\) is the muscle fiber length in the heart, \(v(t)\) is the stimulus, \(\mu > 0\) and \(\alpha\) are parameters. The model (1.5) has an equilibrium point at
\((x \ v)^T = \left(x_0 - \frac{x_0^3}{3} + \alpha x_0\right)^T\). Equation (1.5) is solved numerically using the Matlab function \texttt{ode45} for \(\mu = 10\), \(\alpha = 1\) and \(x_0 = 0\), i.e. the equilibrium point is shifted at the origin. The muscle fiber length \(x(t)\) and the stimulus \(v(t)\) are plotted in Figures 1.4(a)-1.4(b) versus time. Also, \(x(t)\) is plotted versus \(v(t)\) in Fig. 1.4(c). Figures 1.4(a)-1.4(b) show that \(x(t)\) and \(v(t)\) are periodic with time and Fig. 1.4(c) shows that in the transition from diastole (long fibers) to systole (short fibers), the contraction happens slowly at first (this ensures no blood backflow which could damage the heart) but that at high enough stimulus the fibers contract suddenly to push the blood all throughout the body.

1.2 Modeling of Periodic Signals

1.2.1 System Identification

System identification is the field of mathematical modeling of dynamic systems and of signal characteristics using experimental data. Many applications can be found in technical and non-technical areas. In control and system engineering, system identification methods are used to estimate suitable models for design of regulators, a prediction algorithm, or for simulation. In signal processing applications such as communications, geophysical engineering and mechanical engineering, models obtained by system identification are used for spectral analysis, fault detection, linear prediction and other purposes. In non-technical areas such as biology, environmental sciences and econometrics, system identification is used to develop models for increasing knowledge of the identified object, or for prediction and control.

System identification can be divided into two categories: \textit{off-line} identification or \textit{batch} identification, and \textit{on-line} identification or \textit{recursive} identification. Off-line identification means that a batch of data are collected from the system and subsequently - as a separate procedure - this batch of data are used to construct a model. On the other hand, in case the model is needed to support decisions that should be taken during the operation of the system, \(i.e.\) “on-line”, it is necessary to infer the model at the same time as the data are collected. The model is then “updated” at each time instant some new data become available. This means that if there is an estimate \(\hat{\gamma}(t-1)\) based on data up to time \(t-1\), then \(\hat{\gamma}(t)\) is computed by some modification of \(\hat{\gamma}(t-1)\) when the new data becomes available.

Nowadays, there is a quite substantial literature on recursive identification \(\textit{see e.g.} \ [51, 59, 65, 66, 69]\). Different recursive algorithms have been suggested and analyzed to find conditions necessary for the recursive algorithm to perform well. Recursive identification methods have the following general features:

- They are the main part of adaptive systems where the action is taken based on the most recent model.
- They do not occupy much memory space, since not all data are stored.
• They can be easily modified to track time-varying parameters.
• They can be used as a first step in a fault detection algorithm, see [13, 35], which is used to find out if the system characteristics have changed significantly.

1.2.2 The Scope of the Thesis

The study and the analysis of periodic signals is broadly important. Very often, transient changes in these signals carry important information about the system. Many modeling schemes are used in this analysis. Some of these schemes may be constrained by the fact that no prior knowledge about the periodic signal is available. Other schemes, like the ones used in this thesis, can benefit from prior knowledge about the signal waveform. In real-time, a frequency estimation algorithm must be able to detect, estimate and track the changing dynamics of either single or multiple periodic signals. This estimation process may be constrained by the level and type of noise.

The study in this thesis concerns modeling of real-valued periodic signals using nonlinear techniques. The suggested nonlinear techniques in this thesis benefit from prior information about the periodic signal such as being generated from nonlinear ordinary differential equation (ODE) or that its shape resembles known waveforms. Hence, an improvement in the performance of these methods is expected as compared to other methods that do not make use of any priors. This is provided that the assumed model structure is suitable for these periodic signals.

1.3 Previous Approaches to Periodic Signal Modeling

The problem of periodic signal modeling has been approached from many directions. Each one of these approaches has its advantages and disadvantages. In this section, some of these earlier approaches will be discussed. More specifically, periodograms, modeling of line-spectra and adaptive comb filtering will be described. New approaches to the problem of modeling periodic signals will be presented in Sections 1.4-1.5.

1.3.1 The Periodogram

The periodogram should be considered as the first method used to determine possible hidden periodicities in time series. For \( N \) samples of a signal \( y(t) \), the periodogram is defined as

\[
\hat{\phi}_p(\omega) = \frac{1}{N} \left| \sum_{t=1}^{N} y(t) e^{-i\omega t} \right|^2.
\]

Practically, it is not possible to evaluate \( \hat{\phi}_p(\omega) \) over a continuum of frequencies. Hence, the frequency variable must be sampled for the purpose of computing
\( \hat{\phi}_p(\omega) \). The following sampling scheme is most commonly used:

\[
\omega_k = \frac{2\pi}{N} k, \quad k = 0, \cdots, N - 1.
\] (1.7)

A direct evaluation of (1.6) requires about \( N^2 \) complex multiplications and additions (\( N^2 \) flops), which may be a heavy computational burden for large values of \( N \). For this reason, the Fast Fourier Transform (FFT) algorithm, see e.g. [87, 88], is used as a standard technique. The FFT algorithm requires only \( O(N \log N) \) flops.

The periodogram is simple and easy to apply but it has sometimes a poor performance for estimation of the power spectral density (PSD). The following reasons contribute to this fact:

- It is an asymptotically unbiased spectral estimator. In practice, the user may need to increase the number of samples \( N \) too much to reduce the bias significantly.

- It is an inconsistent spectral estimator, *i.e.* it continues to fluctuate around the true PSD with nonzero variance even if \( N \) is increased without bound.

- It is suffering from a leakage problem or transferring of power from the frequency bands that contain most of the power to bands that contain less or no power. This may give an incorrect indication of the existence of harmonic overtones.

The high variance of the periodogram method motivates the development of modified methods that have lower variance, at a cost of reduced resolution. All the refining periodogram methods, like the Blackman-Tukey method, the Bartlett method, the Welch method and the Daniell method are seeking to reduce the variance of the periodogram by smoothing or averaging the periodogram estimates in some way. The problem with these modified methods is that they increase the bias, see [102] for more details.

### 1.3.2 Modeling of Line-Spectra

In many applications, particularly in communications, radar and sonar, signals can be described as

\[
y(t) = x(t) + e(t),
\]

\[
x(t) = \sum_{k=1}^{n} \alpha_k e^{i(\omega_k t + \phi_k)},
\] (1.8)

where \( x(t) \) is the noise free complex-valued sinusoidal signal and \( \{\alpha_k\}, \{\omega_k\} \) and \( \{\phi_k\} \) are its amplitudes, frequencies and phases, respectively. The following assumptions are mathematically convenient:

- \( \alpha_k > 0 \) for \( \omega_k \in [-\pi, \pi] \) to prevent model ambiguities.
• \{\phi_k\} are independent random variables, uniformly distributed on \([-\pi, \pi]\).
• \(e(t)\) is circular white noise with variance \(\sigma^2\), i.e.
  \[
  \mathbb{E}\{e(t) e^*(s)\} = \sigma^2 \delta_{t,s}, \quad \mathbb{E}\{e(t) e(s)\} = 0,
  \tag{1.9}
  \]
  where \(\mathbb{E}\) is the expectation operator.

The covariance function and the PSD of the noisy sinusoidal signal \(y(t)\) can be calculated under the previous assumptions. It follows that
  \[
  \mathbb{E}\{e^{i\phi_p} e^{-i\phi_j}\} = \delta_{p,j}. \tag{1.10}
  \]

Now, let \(x_p(t) = \alpha_p e^{i(\omega_p t + \phi_p)}\),
  \(\tag{1.11}\)
denote the \(p\)th sine wave in \((1.8)\). It follows from \((1.10)\) that
  \[
  \mathbb{E}\{x_p(t) x_j^*(t - k)\} = \alpha_p^2 e^{i\omega_p k} \delta_{p,j}. \tag{1.12}
  \]
Thus the covariance function of \(y(t)\) is
  \[
  r(k) = \mathbb{E}\{y(t) y^*(t - k)\} = \sum_{p=1}^{n} \alpha_p^2 e^{i\omega_p k} + \sigma^2 \delta_{k,0}. \tag{1.13}
  \]
The PSD of \(y(t)\) is given by the Discrete Time Fourier Transform (DFT) of \\(\{r(k)\}\), which is
  \[
  \phi(\omega) = 2\pi \sum_{p=1}^{n} \alpha_p^2 \delta(\omega - \omega_p) + \sigma^2, \tag{1.14}
  \]
where \(\delta(\omega - \omega_p)\) is the Dirac function. The PSD of \(y(t)\) is called a line spectrum because it consists of a constant level equal to the noise power \(\sigma^2\) with \(n\) vertical lines or impulses located at the sinusoidal frequencies \(\{\omega_k\}\) and having amplitudes of \(\{2\pi\alpha_k^2\}\).

In many applications, the parameters of major interest are the locations of the spectral lines or the sinusoidal frequencies. Once \(\{\hat{\omega}_k\}\) are determined, \(\{\hat{\alpha}_k^2\}\) can be obtained by a least squares method from (see [99])
  \[
  \hat{r}(k) = \sum_{p=1}^{n} \alpha_p^2 e^{i\omega_p k} + \text{residuals}, \tag{1.15}
  \]
or both \(\{\hat{\alpha}_k\}\) and \(\{\hat{\phi}_k\}\) can be derived by a least squares method from
  \[
  y(t) = \sum_{k=1}^{n} \beta_k e^{i\omega_k t} + \text{residuals}, \tag{1.16}
  \]
where
  \[
  \beta_k = \alpha_k e^{i\phi_k}. \tag{1.17}
  \]
Many methods have been suggested for solving this frequency estimation problem, for example the nonlinear least squares method, the high order Yule-Walker method, the Min-Norm method, the MUSIC method and the ESPRIT method, see e.g. [102].

The main advantages of the line-spectra approach is that it can be used for high resolution applications as in radar and sonar, where it gives accurate frequency estimates. On the other hand, line-spectra methods usually require a higher computational burden than the refining periodogram methods.

1.3.3 The Adaptive Comb Filter

A comb filter is a filter that has a comb-like frequency response, see [49, 54, 75, 87]. The adaptive comb filtering technique suggested in [75] for modeling periodic signals consists of two cascaded infinite impulse response (IIR) sections. These two sections are only parameterized by the estimated fundamental frequency of the periodic signal. The first section is used to estimate the fundamental frequency and enhance the periodic signal. Given the estimated fundamental frequency, the harmonic amplitudes and phases are estimated separately in the second section.

In order to describe the algorithm presented in [75], let \( x(t) \) be the periodic signal whose parameters are to be estimated. Thus

\[
x(t) = \sum_{k=1}^{n} C_k \sin(k\omega_o t + \phi_k),
\]

where \( \omega_o \) is the fundamental frequency, \( C_k \) and \( \phi_k \) are the amplitude and phase of the \( k \)th harmonic component of \( x(t) \), respectively. The number \( n \) is the assumed number of truncated harmonics in \( x(t) \). In practice, \( n \) is chosen so that the energy in the remaining harmonics is sufficiently small. The remaining harmonics are considered as part of the noise. Hence, the measured signal at time \( t \) is assumed to be

\[
y(t) = x(t) + v(t),
\]

\[
y(t) = \sum_{k=1}^{n} C_k \sin(k\omega_o t + \phi_k) + v(t),
\]

where \( v(t) \) is zero-mean white noise with variance \( \sigma^2 \).

The adaptive comb filtering approach is based on using an IIR whitening filter for \( y(t) \), i.e. the filter that produces "approximately" a white noise if its input is \( y(t) \). For a sum of \( n \) sine waves (not necessarily harmonics) with additive white noise, \( y(t) \) can be shown to satisfy the following autoregressive moving-average (ARMA) model (see [102]):

\[
L(q^{-1}) y(t) = L(q^{-1}) v(t),
\]

where \( L(q^{-1}) \) is the 2\( n \)th-order polynomial in the unit delay operator \( q^{-1} \), whose zeros are on the unit circle at the sine wave frequencies. Since the nulls
of $L(q^{-1})$ are at the sine wave frequencies, it does not follow from (1.20) that $y(t) = v(t)$. Note that the cancellation of $L(q^{-1})$ in (1.20) is not allowed because the zeros of $L(q^{-1})$ are on the unit circle.

Since in this special case, the zeros of $L(q^{-1})$ are at integral multiples of $\omega_o$, $L(q^{-1})$ can be written as

$$L(q^{-1}) = \prod_{k=1}^{n} (1 + \alpha_k q^{-1} + q^{-2}), \quad (1.21)$$

where

$$\alpha_k = -2 \cos(k\omega_o). \quad (1.22)$$

Due to the symmetry of $L(q^{-1})$,

$$L(q^{-1}) = 1 + l_1 q^{-1} + \cdots + l_n q^{-n} + \cdots + l_1 q^{-2n+1} + q^{-2n}. \quad (1.23)$$

The whitening filter of $y(t)$ can be approximated by

$$H(q^{-1}) = \frac{L(q^{-1})}{L(q^{-1})} = \frac{\prod_{k=1}^{n} (1 + \alpha_k q^{-1} + q^{-2})}{\prod_{k=1}^{n} (1 + \rho \alpha_k q^{-1} + \rho^2 q^{-2}).} \quad (1.24)$$

From (1.24), the whitening filter is stable when $\rho < 1$. It is characterized by $2n$ zeros on the unit circle at $\{e^{\pm jk\omega_o}, 1 \leq k \leq n\}$ and $2n$ poles at $\{\rho e^{\pm jk\omega_o}, 1 \leq k \leq n\}$. The parameter $\rho$ is chosen by the user; typical values are $0.95 - 0.995$.

It can be easily observed from (1.20), (1.21) and (1.24) that the error signal

$$\varepsilon(t) = H(q^{-1}) y(t), \quad (1.25)$$

approximates the noise $v(t)$ when $\rho$ is sufficiently close to one and $\alpha_k$ satisfies (1.22).

The unknown parameter vector is given by

$$\theta = (\omega_o \ C_1 \ldots C_n \ \phi_1 \ldots \phi_n)^T. \quad (1.26)$$

A maximum likelihood estimation algorithm of $\theta$ would require a nonlinear search for $2n + 1$ parameters. For simplicity, the algorithm is divided into two cascaded parts, as is illustrated in Fig. 1.5. As shown in the figure, the first part of the algorithm is the recursive prediction error adaptive comb filter (RPEACF), which estimates the fundamental frequency $\omega_o$ and enhances the periodic signal $x(t)$. The second part, based on $\hat{\omega}_o$ and $\hat{x}(t)$, estimates the amplitudes $\{C_k\}$ and the phases $\{\phi_k\}$, after parameter transformation, using a linear recursive least squares (RLS) algorithm, see [75, 99].
The adaptive comb filtering approach has the following advantages:

- It does not need any stability monitoring as do other RPE algorithms, cf. [75].
- It estimates also the amplitudes and the phases of the harmonics.

On the other hand, it misses the following:

- It does not give any information on the underlying nonlinearity or nonlinear dynamics (in cases where a nonlinearity causes the harmonic spectrum).
- It does not benefit from any prior information about the waveform, other than the fact that it is harmonic.
- It is essential for the approach to find a good estimate for the harmonics number $n$. The use of an underestimated $n$ usually adds a distortion to the filtered signal. The use of an overestimated $n$ adds some noise to the output.

The study of this thesis concentrates on nonlinear techniques for the periodic signal modeling problem, namely: periodic signal modeling based on the Wiener model structure and periodic signal modeling using orbits of second-order nonlinear ordinary differential equations (ODE’s). An introduction to these approaches are given in the following sections.

1.4 Periodic Signal Modeling Based on the Wiener Model Structure

Over the years considerable attention has been given to the identification of linear systems, see e.g. [51, 59, 67, 69, 99]. Linear systems have proven their usefulness in numerous engineering applications, and many theoretical results have been derived for the identification and control of these systems. However, many real-life systems inherently show nonlinear dynamic behavior. Consequently, the use of linear models has its limitations. When performance requirements are high, the linear model may not be accurate enough, and nonlinear models may have to be used. Recently, the field of identifying nonlinear systems has attracted an increasing number of researchers.

Next, an introduction to some block-structured models used in nonlinear system identification is given. Also, the original adaptive nonlinear modeler, which introduced the piecewise linear parametrization of static nonlinearities in system identification, is discussed.

1.4.1 Block-Structured Models

Block-structured models are used to model nonlinear systems that can be represented by interconnections of linear dynamics and static nonlinear ele-
ments. There are four commonly used block-structured models in the literature, namely: the Wiener model [11, 53, 77, 83, 117-119], the Hammerstein model [9, 12, 103, 109, 112, 113, 129], the Wiener-Hammerstein cascade model [15, 16, 18, 19, 21, 93] and the Hammerstein-Wiener cascade model [7, 8, 130]. For these nonlinear models, it is assumed that only the input and the output signals of the model are measurable. In the following, the mentioned block-structured models are discussed individually.

**The Wiener Model Structure**

The Wiener model structure consists of a linear dynamic system followed by a static nonlinearity, as shown in Fig. 1.6. The input and output of the total system can be measured, possibly with noise, but not the intermediate signal. Wiener models arise in practice whenever a measurement device has a nonlinear characteristic, see [17, 19, 37, 44, 60, 76, 82].

Consider the following parameteric description for the linear dynamics and the static nonlinearity. Assume that the intermediate signal $y(t)$ is related to the input signal by

$$ y(t) = \frac{B(q^{-1})}{A(q^{-1})} u(t), \quad (1.27) $$

where $A(q^{-1})$ and $B(q^{-1})$ are polynomials in the delay operator $q^{-1}$, given by

$$ A(q^{-1}) = 1 + a_1 q^{-1} + \cdots + a_n q^{-n_a}, $$
$$ B(q^{-1}) = b_0 + b_1 q^{-1} + \cdots + b_n q^{-n_b}. \quad (1.28) $$

Also, let the output of the Wiener model $y_n(t)$ be defined as

$$ y_n(t) = f(y(t)), \quad (1.29) $$

where $f(\cdot)$ is the function of the static nonlinearity. Hence,

$$ y_n(t) = f \left( \frac{B(q^{-1})}{A(q^{-1})} u(t) \right). \quad (1.30) $$

It is not possible to identify the linear dynamics independently of the static nonlinearity. Independent parameterization of the two blocks requires maintaining the static gain to be constant in one of the blocks, see [118, 119]. This is due to the fact that the total input-output static gain of the Wiener model is the product of the static gains of the two cascaded blocks. Also, it is important in what way disturbances enter to the system, see e.g. [119]. As shown in
1. Introduction

Figure 1.7: Two possible cases for a disturbance $w$ to enter a Wiener type system.

Fig. 1.7, in the upper figure $w$ is considered as a measurement noise but in the lower figure $w$ is considered as a disturbance entering into the system.

Using a parametric description for the linear dynamic block and the static nonlinearity, a prediction error criterion, see [99], can be used to estimate the parameters of the Wiener model. Due to complexity of the prediction error criterion, it is usually minimized numerically, e.g. by Gauss-Newton method. Starting with a good initial estimate, the numerical search can be designed to guarantee convergence to a local minimum of the criterion function.

Also, nonparametric approaches have been used to identify the Wiener model in [11, 37]. In [37], the static nonlinearity was assumed to be invertible. Hence, the inverse of the static nonlinearity was expressed as a regression function and a nonparametric regression estimation technique was developed. Also, a method for recovering the impulse response of the linear block was presented. In [11], a nonstandard basis of the Fourier series representation was used as an input to the Wiener model. Then, the relationship between the phase of the linear block and the output was exploited. Hence, the phase of the linear block was extracted based on the discrete Fourier transform (DFT) of the output measurements. Following that step, the static nonlinearity as well as the gain of the linear block was estimated.

Figure 1.8: The Hammerstein model structure.

The Hammerstein Model Structure

The Hammerstein model consists of a static nonlinearity followed by a linear dynamic system, as shown in Fig. 1.8. The Hammerstein model is considered as the easiest nonlinear model to use for identification purposes compared to
other nonlinear model structures. To explain this, assume that the output signal \(y_n(t)\) is given by
\[
y_n(t) = \frac{B(q^{-1})}{A(q^{-1})} y(t),
\]
(1.31)
where \(A(q^{-1})\) and \(B(q^{-1})\) are given by (1.28). Also, in this case the output of the static nonlinearity \(y(t)\) is given by
\[
y(t) = f(u(t)),
\]
(1.32)
where \(f(\cdot)\) is the function of the unknown static nonlinearity. Approximating \(y(t)\) as
\[
\tilde{y}(t) = \sum_{i=1}^{n} \beta_i f_i(u(t))
\]
(1.33)
where \(\{\beta_i\}_{i=1}^{n}\) are unknown constants and \(\{f_i(\cdot)\}_{i=1}^{n}\) are suitable and known basis functions. In this case, it is possible to use independent parameterization of the two blocks by redefining the input as
\[
\tilde{u}(t) = \left( f_1(u(t)) \quad f_2(u(t)) \quad \cdots \quad f_n(u(t)) \right),
\]
(1.34)
i.e. transforming the Hammerstein model to a linear MISO model. This transformation allows for the identification of Hammerstein models using linear techniques such as the least squares (LS) algorithm.

One can find a quite substantial literature dealing with the identification of Hammerstein models. Generally speaking, existing identification methods for Hammerstein models can be divided into iterative methods, see e.g. \([101, 112, 113]\), over-parameterization methods, see e.g. \([22]\), stochastic methods, see e.g. \([16, 38]\), separable least squares methods, see e.g. \([9, 116]\), blind identification methods, see e.g. \([12]\) and frequency domain methods, see e.g. \([10]\).

Figure 1.9: The Wiener-Hammerstein cascade model.

### The Wiener-Hammerstein Model Structure

Identification of the Wiener-Hammerstein model structure is a more difficult problem compared to identifying the Wiener and the Hammerstein models. To give an idea about the difficulty of identifying the Wiener-Hammerstein model, assume that the output signal \(y_n(t)\) is given by (see Fig. 1.9)
\[
y_n(t) = \frac{B(q^{-1})}{A(q^{-1})} y_2(t),
\]
(1.35)
where \(A(q^{-1})\) and \(B(q^{-1})\) are given by (1.28), and
\[
y_2(t) = f(y_1(t)).
\]
(1.36)
1. Introduction

Also, assume

\[ y_1(t) = \frac{D(q^{-1})}{C(q^{-1})} u(t), \]  

(1.37)

where

\[ C(q^{-1}) = 1 + c_1 q^{-1} + \cdots + c_n q^{-nc}, \]
\[ D(q^{-1}) = d_0 + d_1 q^{-1} + \cdots + d_n q^{-nd}. \]  

(1.38)

Now, the output of the Wiener-Hammerstein model is

\[ y_n(t) = B(q^{-1}) A(q^{-1}) f \left( \frac{D(q^{-1})}{C(q^{-1})} u(t) \right). \]  

(1.39)

In this case, it is more difficult, compared to the Wiener and the Hammerstein models, to obtain an efficient algorithm to estimate the two linear dynamic blocks and the static nonlinearity simultaneously. In [16, 18, 19], correlation methods were used in order to identify systems described by the Wiener-Hammerstein model. Another approach for the identification of the Wiener-Hammerstein model was suggested in [126]. It consists of estimating, in the SISO case, impulse responses of the linear subsystems and the parameters of the nonlinear element. In [21], a technique for recursive identification of Wiener-Hammerstein model with extension to the MISO case was presented. The technique uses a transformation which leads to a unique and equivalent realization of the equations (1.35)-(1.38). After that, a weighted extended least squares (WELS) method, see [69], was employed to estimate recursively and separately the parameters of the linear subsystems and the static nonlinear element.

![Figure 1.10: The Hammerstein-Wiener cascade model.](image)

The Hammerstein-Wiener Model Structure

Recently, the problem of identifying the Hammerstein-Wiener model was considered, see [7, 8, 130]. The Hammerstein-Wiener model consists of a linear block embedded between two static nonlinear blocks, see Fig. 1.10. In process control environments, the Hammerstein-Wiener model can be motivated by considering the input nonlinear block as the actuator nonlinearity and the output nonlinear block as the process nonlinearity. The Hammerstein-Wiener model structure can be considered as a Wiener model structure with an artificial input \( \bar{u}(t) \), see Eqs. (1.33)-(1.34).

In [7], an over-parameterization method was introduced. The method of [7] suggests a two-stage identification scheme based on the assumption that the two nonlinear blocks are a priori known smooth nonlinear functions. In [8],
the previous assumption was relaxed and a blind identification technique was introduced to estimate the linear block and the two unknown static nonlinearities. In [130], a relaxation algorithm which is numerically simpler and more reliable than general nonlinear search algorithms was presented.

1.4.2 The Adaptive Nonlinear Modeler

One of the main objectives in identification of block-structured models is to identify the static nonlinearity simultaneously with the linear dynamics. In order to overcome the problem that the static nonlinearity is totally unknown, a parameterization of the static nonlinearity is needed. Many different possibilities have been suggested for this purpose, see [44, 95]:

- Power series [30].
- Chebyshev polynomials [30].
- Splines or piecewise polynomials [4, 31, 91, 117].
- Neural networks [50, 52].
- Hinging hyperplanes [23, 89].
- Wavelets [28, 71].

In this thesis, the static nonlinearity is parameterized by piecewise polynomials and assumed to be single-valued on each interval where the slope of the driving signal has a constant sign. Situations that include multi-valued static nonlinearities, like hysteresis and backlash, will not be considered. The parameters of the piecewise polynomials are chosen as the values of the estimated nonlinear output in some user chosen grid points exactly as done in [4, 117].

The static nonlinearity output \( y \) is modeled as

\[
y = f(x),
\]

where \( x \) is the input. The model is parameterized in terms of the values of the function \( f \) in some user chosen grid points \( x_1, \ldots, x_n \), see Fig. 1.11. Different kinds of interpolation can be used to evaluate the nonlinear function values at intermediate points. Thus, the parameter vector of the static nonlinearity is here defined as

\[
\theta = \begin{pmatrix} f(x_1) & f(x_2) & \cdots & f(x_n) \end{pmatrix}^T.
\]

Assuming linear interpolation, the static nonlinearity model becomes (in \([x_k, x_{k+1}]\))

\[
f(x) = \frac{x_{k+1} - x}{x_{k+1} - x_k} f(x_k) + \frac{x - x_k}{x_{k+1} - x_k} f(x_{k+1}).
\]
of the static nonlinearity becomes (in $[x_k, x_{k+1}]$)

$$f(x) = f(x_{k+\frac{1}{2}}) + \frac{f(x_{k+1}) - f(x_k)}{(x_{k+1} - x_k)} (x - x_{k+\frac{1}{2}}) + 2 \left( f(x_{k+1}) - 2f(x_{k+\frac{1}{2}}) + f(x_k) \right) \frac{1}{(x_{k+1} - x_k)^2} (x - x_{k+\frac{1}{2}})^2. \quad (1.43)$$

**1.4.3 The Periodic Signal Modeling Approach Based on the Wiener Model**

The approach of periodic signal modeling based on the Wiener model structure is the main theme of the first part of this thesis. The approach was introduced in [121]. Also, an on-line tracking scheme based on this approach was presented in [47].

The suggested method relies on the fact that a sine wave that is passed through a static nonlinear function produces a harmonic spectrum of overtones. Consequently, the estimated signal model can be parameterized as a driving periodic wave with unknown period in cascade with a piecewise linear function. The driving wave can be chosen depending on any prior knowledge. If, for example, the signal is known to be close to a triangle wave, this can be readily exploited by the proposed method. The driving frequency and the parameters of the nonlinear output function are estimated simultaneously. Hence, a periodic wave with unknown fundamental frequency in cascade with a parameterized and unknown nonlinear function can be used as a signal model for an arbitrary periodic signal, as shown in Fig. 1.12.

A recursive prediction error (RPE) algorithm is used to estimate recursively the periodic function frequency, which represents the fundamental frequency of
the true periodic wave. The algorithm estimates, simultaneously with the fundamental frequency, the static nonlinearity parameterized in some grid points chosen by the user.

1.5 Periodic Signal Modeling Using Orbits of Second-Order Nonlinear ODE’s

The study of limit cycle oscillations is an important topic in the engineering field. Limit cycle oscillations problem encountered in many applications such as aerodynamics, see [6], combustion systems, see [26], relay feedback systems, see [56], reaction kinetics, queuing systems and oscillating chemical reactions, see [14]. Hence, mathematical modeling of limit cycles becomes an essential topic that helps to better understand and/or to avoid limit cycle oscillations in different fields.

Figure 1.12: The approach to periodic signal modeling based on the Wiener model structure. Note that the signal generation within the dashed frame may also be based on other structures. As long as the measured signal $y$ is periodic, the method of the thesis applies.
The second major nonlinear technique presented in this thesis for modeling periodic signals is based on using orbits of second-order nonlinear ODE’s. In this section a background for this approach is given.

1.5.1 Dynamical Systems

The dynamics of any situation refers to how the situation changes over the course of time. A dynamical system is a mathematical description for how the system setting changes or evolves from one moment of time to the next. Examples of dynamic systems include:

- The solar system.
- The weather.
- The economy.
- The human body (heart, brain, lungs, ...).
- Ecology (plant and animal populations).
- Chemical reactions.
- The electrical power grid.
- The internet.

1.5.2 Nonlinear Models of Dynamical Systems

Large classes of dynamical systems can be modeled by a finite number of coupled first-order ordinary differential equations, see \[ [40, 57, 84, 131] \],

\[
\begin{align*}
\dot{x}_1 &= f_1(t, x_1, \ldots, x_n, u_1, \ldots, u_p), \\
\dot{x}_2 &= f_2(t, x_1, \ldots, x_n, u_1, \ldots, u_p), \\
& \quad \vdots \\
\dot{x}_n &= f_n(t, x_1, \ldots, x_n, u_1, \ldots, u_p),
\end{align*}
\]

(1.44)

where \( x_1, x_2, \ldots, x_n \) are the state variables which represent the memory that the dynamical system has of its past. In (1.44) \( \dot{x}_i \) denotes the derivative of \( x_i \) and \( u_1, u_2, \ldots, u_p \) are the inputs to the dynamical system.

A vector notation is usually used to write (1.44) in a compact form. To do so, define

\[
\begin{align*}
x &= \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix}^T, \\
u &= \begin{pmatrix} u_1 & u_2 & \cdots & u_p \end{pmatrix}^T, \\
f(t, x, u) &= \begin{pmatrix} f_1(t, x, u) & f_2(t, x, u) & \cdots & f_n(t, x, u) \end{pmatrix}^T.
\end{align*}
\]
Now, the $n$ first-order differential equations (1.44) can be rewritten as one $n$-dimensional first-order vector differential equation

$$\dot{x} = f(t, x, u).$$

Equation (1.45) is called the state equation referring to $x$ as the state and $u$ as the input. Sometimes, Eq. (1.45) is associated with the following output equation

$$y = h(t, x, u).$$

Equation (1.46) defines a $q$-dimensional output vector $y$ that comprises variables of particular interest of the analysis of the dynamical system, e.g. variables that can be physically measured or variables that are required to behave in a specific manner. It is usually referred to (1.45) and (1.46) together as the state-space model.

In many situations the analysis of dynamical systems deals with the state equation without explicit presence of an input $u$. In this case the state equation becomes

$$\dot{x} = f(t, x).$$

Equation (1.47) is called an unforced state equation. Working with unforced state equations does not necessarily mean that the input to the system is zero. It could be that the input has been specified as a given function of time, a given feedback function of the state, or both.

A special case of (1.47) arises when the function $f$ does not depend explicitly on $t$, i.e.

$$\dot{x} = f(x).$$

In this case the system is said to be autonomous or time-invariant. The behavior of an autonomous system is invariant to shifts in the time origin, since changing the time variable from $t$ to $\tau = t - \Delta t$ does not change the right-hand side of the state equation. If the system is not autonomous, then it is called non-autonomous or time-varying.

### 1.5.3 Solutions of Ordinary Differential Equations

In order to use the state equation defined by (1.47) as a useful mathematical model of different physical systems, a solution for Eq. (1.47) must exist. Also, Eq. (1.47) must be able to predict the future state of the system from its current state at $t_0$. This means that the initial-value problem

$$\dot{x} = f(t, x), \quad x(t_0) = x_0,$$

must have a unique solution. It is shown in the literature, see e.g [62, 92], that existence and uniqueness of solutions to Eq. (1.49) can be ensured by imposing some constraints on the right hand side function $f(t, x)$. These constraints are:

- The function $f(t, x)$ is piecewise continuous in time $t$. 

• There is a constant \( L \geq 0 \) such that the function \( f(t, x) \) satisfy the Lipschitz condition given by

\[
\| f(t, x) - f(t, y) \| \leq L \| x - y \|,
\]

for all \( x \) and \( y \) in some neighborhood of \((t_0, x_0)\).

### 1.5.4 Second-Order Autonomous Systems

Second-order autonomous systems occupy an important place in the study of nonlinear systems because solution trajectories can be represented by curves in the plane, see [62, 92, 111]. This allows for easy visualization of the qualitative behavior of the system. A second-order autonomous systems are represented by two scalar differential equations

\[
\begin{align*}
\dot{x}_1 &= f_1(x_1, x_2), \\
\dot{x}_2 &= f_2(x_1, x_2),
\end{align*}
\]

or in a more compact form as given in (1.48), where

\[
\begin{align*}
\dot{x}(t) &= \left( \begin{array}{c}
\dot{x}_1(t) \\
\dot{x}_2(t)
\end{array} \right), \\
f(x) &= \left( \begin{array}{c}
f_1(x_1, x_2) \\
f_2(x_1, x_2)
\end{array} \right).
\end{align*}
\]

Let \( x(t) = \left( \begin{array}{c} x_1(t) \\ x_2(t) \end{array} \right)^T \) be the solution of (1.51) that starts at a certain initial state \( x_0 = \left( \begin{array}{c} x_{10} \\ x_{20} \end{array} \right)^T \). The locus in the \( x_1-x_2 \) plane of the solution \( x(t) \), for all \( t \geq 0 \), is a curve that passes through the point \( x_0 \). This curve is called a trajectory or orbit. The \( x_1-x_2 \) plane is usually called state plane or phase plane.

### 1.5.5 Limit Cycles

Oscillation is one of the most important phenomena that occur in dynamical systems. In linear systems, sustained oscillations result from oscillatory inputs and their amplitude and frequency are uniquely dependent on initial conditions of the states of the system. For nonlinear systems, periodic oscillations may arise from non-oscillatory inputs. There are nonlinear systems that can go into an oscillation of fixed amplitude and frequency, irrespective of the initial state. Sustained periodic oscillations in nonlinear systems are called limit cycles, see [41, 45, 104, 125, 128].

A system oscillates when it has a nontrivial periodic solution

\[
x(t + T) = x(t), \quad \forall \ t \geq 0,
\]

for some \( T > 0 \). The word “nontrivial” is used to exclude constant solutions corresponding to equilibrium points. The image of the periodic solution in the phase plane is a closed trajectory, which is usually called a periodic orbit.
1.5.6 Stability of Periodic Solutions

A periodic solution is (asymptotically) stable if its associated periodic orbit is also (asymptotically) stable. The stability of a periodic orbit can be exploited from the concept of stability of invariant sets. This is because the periodic orbit is considered as a special case of invariant sets. The issue of stability of periodic orbits is discussed in detail in [62]. A brief discussion is only introduced in this section.

Let $\mathcal{M}$ be a closed invariant set of the second-order autonomous system (1.51). Define an $\epsilon$-neighborhood of $\mathcal{M}$ by

$$
\mathcal{V}_\epsilon = \{ x \in \mathbb{R}^2 \mid \text{dist}(x, \mathcal{M}) < \epsilon \},
$$

(1.53)

where \( \text{dist}(x, \mathcal{M}) \) is the minimum distance from \( x \) to a point in \( \mathcal{M} \), i.e.

$$
\text{dist}(x, \mathcal{M}) = \inf_{y \in \mathcal{M}} \| x - y \|.
$$

(1.54)

Hence, the closed invariant set $\mathcal{M}$ of (1.51) is

- **stable** if, for each $\epsilon > 0$, there is $\delta > 0$ such that

$$
x(0) \in \mathcal{V}_\delta \Rightarrow x(t) \in \mathcal{V}_\epsilon, \ \forall t \geq 0.
$$

(1.55)

- **asymptotically stable** if its is stable and $\delta$ can be chosen such that

$$
x(0) \in \mathcal{V}_\delta \Rightarrow \lim_{t \to \infty} \text{dist}(x, \mathcal{M}) = 0.
$$

(1.56)

**Example 1.3.** A periodic orbit.

Consider the following second-order nonlinear system

$$
\begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= -x_1 + (1 - x_1^2 - x_2^2)x_2.
\end{align*}
$$

(1.57)

By inspection, it is seen that (1.57) admits a periodic solution

$$
\gamma_p(t) := \begin{cases} 
  x_1 = \sin t \\
  x_2 = \cos t, & t \in [0, 2\pi].
\end{cases}
$$

(1.58)

The phase portrait of this periodic solution for different initial states is given in Fig. 1.13.

The associated periodic orbit with $u_p(t)$ is defined by

$$
\gamma_p = \{ x \in \mathbb{R}^2 \mid r = 1 \}, \ \text{where} \ r = \sqrt{x_1^2 + x_2^2}.
$$

The $\mathcal{V}_\epsilon$ neighborhood of $\gamma_p$ is defined in this case by the annular region

$$
\mathcal{V}_\epsilon = \{ x \in \mathbb{R}^2 \mid 1 - \epsilon < r < 1 + \epsilon \}. 
$$
Thus, it can be noticed that given $\epsilon > 0$, we can choose a value for $\delta$ such that conditions (1.55)-(1.56) are satisfied. This means that any solution starting in the $V_\delta$ neighborhood will asymptotically go to $\gamma_p$ which can be noticed from Fig. 1.13, i.e. $\gamma_p$ is asymptotically stable.

The previous conclusion can be also investigated in the sense of Lyapunov stability, see [62]. Consider the Lyapunov function

$$V(x) = (1 - x_1^2 - x_2^2)^2, \quad x_1^2 + x_2^2 \neq 1. \quad (1.59)$$

Differentiating both sides of (1.59) w.r.t. time $t$ and using (1.57) give

$$\dot{V}(x) = -4(1 - x_1^2 - x_2^2)[x_1 \dot{x}_1 + x_2 \dot{x}_2]$$

$$= -4x_2^2(1 - x_1^2 - x_2^2)^2 \leq 0. \quad (1.60)$$

Also, using LaSalle’s theorem, see [62], $\dot{V}(x) = 0$ only at the origin (for $x_1^2 + x_2^2 \neq 1$). Hence, the periodic solution of (1.57) is asymptotically stable.

### 1.5.7 The Periodic Signal Modeling Approach Using Periodic Orbits

Many systems that generate periodic signals can be described by second-order nonlinear ODE’s with polynomial right hand sides. Examples include tunnel diodes, pendulums, negative-resistance oscillators and biochemical reactors, see [57, 62]. Therefore, by using a second-order nonlinear ODE model for the periodic signal, it can be expected that there are good opportunities to obtain highly accurate models by only estimating a few parameters.

It is proved in [122, 123] that a second-order nonlinear ODE is sufficient to model a large class of periodic signals provided that the phase plane plot
1.5. Periodic Signal Modeling Using Orbits of Second-Order Nonlinear ODE’s

that is constructed from the periodic signal and its first derivative lacks any intersections or limiting cases such as corners, stops and cusps. Intersected phase plots need higher order nonlinear ODE’s to be modeled accurately. The results of [122, 123] are discussed in more details in Chapter 6.

The idea of the suggested approach for modeling periodic signals using second-order nonlinear ODE model is based on the following steps:

- **Model structure:**
  The periodic signal is modeled as a function of the state of the following second-order nonlinear ODE
  \[
  \ddot{y}(t) = f(y(t), \dot{y}(t), \theta),
  \]
  where \( \theta \) is an unknown parameter vector to be determined. Hence, choosing
  \[
  \begin{pmatrix}
  x_1 \\
  x_2
  \end{pmatrix} = \begin{pmatrix}
  y(t) \\
  \dot{y}(t)
  \end{pmatrix},
  \]
  the model given in (1.61) is now transferred to the following state space model
  \[
  \begin{pmatrix}
  \dot{x}_1 \\
  \dot{x}_2
  \end{pmatrix} = \begin{pmatrix}
  x_2(t) \\
  f(x_1(t), x_2(t), \theta)
  \end{pmatrix},
  \]
  \[
  y(t) = \begin{pmatrix}
  1 & 0
  \end{pmatrix} \begin{pmatrix}
  x_1(t) \\
  x_2(t)
  \end{pmatrix}.
  \]

- **Model parameterization:**
  The function of the right hand side of the second state equation of (1.63), is then parameterized by expanding it in terms of truncated basis functions selected by the user. In case polynomial basis are considered, as in this thesis, the right hand side of the second state equation of (1.63) becomes
  \[
  f(x_1(t), x_2(t), \theta) = \sum_{l=0}^{L} \sum_{m=0}^{M} \theta_{l,m} x_1^l(t)x_2^m(t).
  \]
  In this case, the parameter vector \( \theta \) takes the form
  \[
  \theta = \begin{pmatrix}
  \theta_{0,0} & \cdots & \theta_{0,M} & \cdots & \theta_{L,0} & \cdots & \theta_{L,M}
  \end{pmatrix}^T.
  \]

- **Model discretization:**
  The ODE model is then discretized in time to allow implementation of different algorithms.

- **Parameter vector estimation:**
  Different statistical algorithms are then developed to estimate the parameter vector \( \theta \).
1.6 Why and When to Use the Proposed Methods in this Thesis?

A direct and important question that may arise in the mind of the reader of this thesis is: why and when to use the approaches introduced in this thesis? The following two examples aim at answering that question. More detailed answers will follow in Part III of this thesis.

Example 1.4. Comparison between some existing methods for modeling periodic signals and the approach of Part I.

In order to compare the performance of the approach of modeling periodic signals based on the Wiener model structure with the periodogram approach and the line-spectra approach, the following simulations were performed.

The data were generated according to the following description: the driving wave was given by $u(t, \theta_1) = X^0 \sin \omega^o t$ where $\theta_1 = (X^0 \quad \omega^o)^T = (1 \quad 2\pi \times 0.05)^T$. Two static nonlinearities were used as shown in Fig. 1.14. The grid points and the parameters of the static nonlinearities were:

$$g_1 = \begin{pmatrix} -1 & -0.3 & -0.15 & 0.15 & 0.3 & 1 \end{pmatrix}^T,$$

$$g_2 = \begin{pmatrix} -1 & -0.3 & 0.3 & 1 \end{pmatrix}^T,$$

$$\theta_1 = \begin{pmatrix} -0.8 & -0.3 & 0.3 & 0.8 \end{pmatrix}^T, \quad u(t, \theta_1) \in I_1,$$

$$\theta_2 = \begin{pmatrix} -0.8 & -0.5 & 0.5 & 0.8 \end{pmatrix}^T, \quad u(t, \theta_2) \in I_2.$$

where $u(t, \theta_1) \in I_1$ for positive slopes and $u(t, \theta_2) \in I_2$ for negative slopes.
1.6. Why and When to Use the Proposed Methods in this Thesis?

respectively. The reason for using two static nonlinearities is to generate un-
symmetric nonlinear distorted periodic signal which is a common situation in
practice, see Remark 2.1 for details. The generated data are shown in Fig. 1.15.

The fixed grid point algorithm of Chapter 3, the ESPRIT method, see [102],
and the periodogram method, see Section 1.3.1, were used to model the data.
The model from the ESPRIT method was evaluated by estimating 12 frequency
components and the amplitudes and the phases of these components were esti-
mated using the linear least squares method. The model from the periodogram
method was evaluated by the extraction of the 12 most powerful frequency
bins and neglecting the rest that consequently represent the model error. The
idea is thus to compare the methods using a fixed number of parameters. The
mean square error (MSE) was calculated as the average value of the squared
modeling error. The results for different number of samples and for different
signal to noise ratios (SNRs) are plotted in Fig. 1.16.

It can be concluded from Fig. 1.16 that the fixed grid point algorithm gives
the lowest MSE for different SNRs. The periodogram method gives a consider-
ably lower MSE than the ESPRIT method for high SNR and a large number of
samples. Also, it can be concluded from the results that the ESPRIT method
may not be suitable for the case of modeling a large number of overtones. The
reason for this worse performance of the ESPRIT method may be caused by
adding the rest of the overtones to the white noise. Since the model of all the
line-spectra methods are based on assuming an additive white noise, this will
not be satisfied in this case. To support this claim, the results obtained by the
ESPRIT method from the original data were compared to the results obtained
by the ESPRIT method from 12 pure sinusoids in additive white noise. The
12 sinusoids were estimated from the original data. The results for both cases
are given in Fig. 1.17. As shown in Fig. 1.17, the MSE for the case obeying
the model assumption is decreasing as the number of samples increases.

The simulation results and the previous discussion indicate that the pro-
posed method is a good choice to be used in cases when the data generation
resembles the imposed model structure.
Example 1.5. Comparison between some existing methods for modeling periodic signals and the approach of Part II.

Similarly to Example 1.4, the approach of periodic signal modeling using second-order nonlinear ODE’s is compared in this example with the periodogram and the ESPRIT method. The data were generated from the Van der Pol oscillator, see [62], given by

\[
\begin{align*}
\dot{x}_1 &= x_2 , \\
\dot{x}_2 &= -x_1 + 2(1 - x_1^2)x_2 .
\end{align*}
\]

(1.67)

The Matlab routine \texttt{ode45} was used to solve (1.67). The initial state of (1.67) was selected as \((x_1(0) \quad x_2(0))^T = (2 \quad 0)^T\). The measured signal was in all examples selected as the first state with white Gaussian noise added. A data
1.6. Why and When to Use the Proposed Methods in this Thesis?

The ESPRIT method for the real data (dash-dot) and for 12 pure sinusoids (solid) [ SNR = 40 dB ].

Figure 1.17: The ESPRIT method for the real data (dash-dot) and for 12 pure sinusoids (solid) [ SNR = 40 dB ].

Figure 1.18: The data generated from the Van der Pol oscillator of Eq. (1.67).

length of $10^4$ samples was generated from (1.67) with a sampling period $h = 0.1$ second, see Fig. 1.18. The period of the generated signal was approximately 7.5 seconds. The least squares algorithm using the automated spectral analysis technique (LS-ASA) of Chapter 10 was used to estimate the parameters of the second-order ODE model with second-degree polynomials ($L = M = 2$).

As done in Example 1.4, the model from the ESPRIT method was evaluated by estimating 10 frequency components and the amplitudes and the phases of these components were estimated using the linear least squares method. The model from the periodogram method was evaluated by the extraction of the 10 most powerful frequency bins and neglecting the rest that consequently represent the model error. The MSE was calculated as the average value of the squared modeling error for the three approaches. The results for data lengths of $10^3$ and $10^4$ samples and for different SNRs are plotted in Fig. 1.19.

It can be concluded from Fig. 1.19 that the approach of second-order non-linear ODE’s gives a more accurate signal model compared to the periodogram and the ESPRIT method especially at moderate and high SNRs.
1.7 Thesis Outline and Contributions

The thesis consists of three parts. The first part provides two new techniques for the periodic signal modeling problem based on the Wiener model structure, and comprises Chapters 2-4. The second part treats the approach of periodic signal modeling using orbits of second-order nonlinear ODE’s. The material of the second part can be found in Chapters 5-12. The third part of this thesis consists of two chapters. Different user aspects are discussed in Chapter 13. In Chapter 14, some comments on different approaches for the periodic signal modeling problem, discussed in Chapter 1 and introduced in this thesis, are given. Also, some topics for future research are introduced.

Chapter 2

In this chapter, the algorithm of [121] is studied by numerical examples to indicate the ability of the algorithm to converge to the true parameter vector. Also, another version of the algorithm is introduced using a piecewise quadratic approximation for the static nonlinearity. Moreover, the algorithm is modified to increase the ability to track fundamental frequency variations. This chapter is based on the paper

Chapter 3

A modified recursive prediction error (RPE) algorithm is introduced in this chapter. The modifications are obtained by introducing an interval in the nonlinear block with fixed static gain. A convergence analysis is performed for the modified algorithm. Also the Cramér-Rao bound (CRB) is derived for the suggested algorithm. The results of this chapter can be found in


Chapter 4

In this chapter, the idea is to give the algorithm of Chapter 3 the ability to estimate the driving frequency and the parameters of the nonlinear output function parameterized in a number of adaptively estimated grid points. This is shown to reduce modeling errors since it gives the algorithm more freedom to choose the suitable grid points. Parts of the material in this chapter can be found in


Chapter 5

Chapter 5 presents the second-order nonlinear ODE model suggested to model periodic signals. Assumptions on the measurements are given. Different model structures, model parameterizations and model discretizations are discussed.

Chapter 6

Sufficiency of the second-order nonlinear ODE model suggested in Chapter 5 to model many periodic signals is addressed in Chapter 6. The chapter presents the main results introduced in [122, 123].

Chapter 7

In this chapter the least squares (LS) estimation algorithm is developed for modeling periodic signals using second-order nonlinear ODE models. The algorithm is studied in case of using finite difference approximations for the derivatives of the modeled signal. This usually leads to biased estimates. To reduce this bias, a Markov estimation algorithm is presented. The chapter is based on the publications

T. Wigren, E. Abd-Elrady and T. Söderström. “Least squares harmonic signal analysis using periodic orbits of ODE’s.” In *Proc. of
1. Introduction


Chapter 8

Chapter 8 introduces two recursive algorithms based on the Kalman filter (KF) and the extended Kalman filter (EKF). The Kalman filter estimate is biased but can be used to generate initial conditions for more elaborate nonlinear techniques such as the EKF. The EKF algorithm shows a superior performance than the KF algorithm. This chapter is based on the paper


Chapter 9

In this chapter, a maximum likelihood (ML) algorithm is developed for estimating the parameter vector responsible for the generation of the modeled signal. The ML estimates are compared statistically to the Cramér-Rao bound, which is also derived for the nonlinear ODE model in this chapter. The results of this chapter can be found in


Chapter 10

In this chapter the spectral analysis technique introduced in [94] is used to estimate the period of the modeled signal in addition to some basic Fourier series coefficients. This allows an evaluation of more accurate estimate for the modeled signal and its derivatives needed to find a LS estimate for the ODE model. The spectral analysis technique of [94] improves the LS estimates significantly. This chapter is based on the publications


Chapter 11

In this chapter Liénard’s equation, which has been used to describe a number of oscillating electrical and mechanical systems, is used as the generating model for the periodic signal. The parameterization in this case makes use of the appropriate conditions imposed on the polynomials of Liénard’s equation to guarantee the existence of a unique and stable periodic orbit. Chapter 11 is based on the paper


Chapter 12

In this chapter a bias analysis for the LS algorithm developed in Chapter 7 is presented. Also, discretization errors are evaluated using Taylor series expansions. Chapter 12 is based on the publications


E. Abd-Elrady and T. Söderström. “Bias analysis in periodic signals modeling using nonlinear ODE’s.” Submitted to 16th IFAC World Congress on Automatic Control, Prague, Czech Republic, July 3-8, 2005

Chapter 13

Different user aspects for the two suggested approaches in this thesis for modeling periodic signals are given in this chapter.

Chapter 14

Comments on different approaches to the periodic signal modeling problem and some topics for future research are given in this chapter.
Part I

Periodic Signal Modeling Based on the Wiener Model Structure
Chapter 2

Periodic Signal Modeling Using Adaptive Nonlinear Function Estimation

2.1 Introduction

The problem of modeling periodic signals has received a great deal of attention in the literature, see e.g. [49, 75]. The algorithm of [121], that is studied here, has the property to give information on the underlying nonlinearity, in cases where the overtones are generated by nonlinear imperfections in the system. In some cases it may also be known that the modeled signal is closer to other signals than to sine waves. This fact can be exploited in the approach of [121]. Therefore, the algorithm of [121] may then be more efficient than methods such as [49, 75] that do not consider such priors.

The method of [121] is based on the fact that a sine wave passing through a static nonlinear function produces a harmonic spectrum of overtones. Hence, the Wiener model structure, cf. Section 1.4.1, is used as a signal model for an arbitrary periodic signal, as shown in Fig. 1.12. The linear block of the Wiener model is a periodic function generator with a tunable fundamental frequency. The periodic function generator is cascaded by a parameterized and unknown static nonlinearity which consists the nonlinear block of the Wiener model.

In this chapter, the nonlinearity is chosen to be piecewise linear exactly as in [121], with the estimated parameters being the function values in a set of user chosen grid points. Also, the nonlinearity can be parameterized in terms of a piecewise quadratic function. In this case, slight modifications in the algorithm of [121] are needed.

The performance of the proposed recursive Gauss-Newton prediction error method (RPEM) for joint estimation of the driving frequency and the parameters of the nonlinear output function is studied by numerical examples. Furthermore, modifications that improve the tracking performance are suggested.

The chapter is organized as follows. In Section 2.2, a review of the algorithm introduced in [121] is given. A parameterization of the nonlinearity as a piecewise quadratic function is presented in Section 2.3. Numerical examples and conclusions are given in Section 2.4 and Section 2.5, respectively.
2.2 Review of the Algorithm of [121]

The method of [121] utilizes a known periodic function as an input to an unknown static nonlinearity. The chosen periodic signal reflects any prior knowledge that is available about the modeled signal. Hence, let the driving input signal \( \hat{u}(t, \omega) \) to be defined as

\[
\hat{u}(t, \omega) = \Lambda(\omega t),
\]

where \( t \) is discrete time and \( \omega \) is the unknown normalized angular frequency. The fact that \( \Lambda(\cdot) \) is periodic means Assumption 2.1.

\[
\Lambda(\omega(t + \frac{2k\pi}{\omega})) = \Lambda(\omega t), \quad k \in \mathbb{Z}.
\]

The method of [121] proceeds by dividing one complete period of \( \Lambda(\omega t) \) into \( L \) disjoint intervals \( I_j, j = 1, \cdots, L \); where \( \Lambda(\cdot) \) satisfies the following condition Assumption 2.2.

\[
\Lambda(\omega t) \quad \text{is a monotone function of} \quad \omega t \quad \text{on each} \quad I_j, j = 1, \cdots, L.
\]

Similarly, the static nonlinear block is divided into \( L \) static nonlinearities \( \{f_j(\theta_j, \Lambda(\omega t))\}_{j=1}^L \) that correspond to these \( L \) disjoint intervals \( I_j \). Hence, the model output is defined as

\[
\hat{y}(t, \omega, \theta_n) = f_j(\theta_j, \Lambda(\omega t)), \quad \Lambda(\omega t) \in I_j, \quad j = 1, \cdots, L, \quad \theta_n = (\theta_1^T \cdots \theta_L^T)^T. \quad (2.2)
\]

Remark 2.1. Assumption 2.2 is introduced in [121] to make the suggested modeling approach more general. This can be explained here as done in [121] by assuming that one static nonlinearity is considered and the driving input periodic function \( \Lambda(\omega t) \) is chosen to be a sinusoid with amplitude \( \alpha \), i.e. \( \Lambda(\omega t) = \alpha \sin(\omega t) \). In this case, \( \hat{y}(t, \omega, \theta_1) = f_1(\theta_1, \alpha \sin(\omega t)) \). If the unknown parameter vector \( \theta_1 \) of the static nonlinearity converged to a fixed parameter vector \( \theta_1^* \), it follows that \( f_1(\theta_1^*, \alpha \sin(\omega t)) = f_1(\theta_1^*, \alpha \sin(\omega/2 - t)) \forall t \). This means that the model output signal in half of the time period, i.e. \( \pi/\omega \), will be given by the model output signal in the second half. In this case, the approach will only be suitable for modeling periodic signals with a symmetric nonlinear distortion.

Remark 2.2. Note that the notation \( \Lambda(\omega t) \in I_j \) - which is used all over Part I of the thesis - means that the phase \( \omega t \) of the periodic function \( \Lambda(\omega t) \) is such that \( I_j \) is in effect. This notation is used intentionally to highlight the switching between static nonlinearities in different intervals.

Considering the use of a piecewise linear model for the parameterization of \( f_j(\theta_j, \Lambda(\omega t)) \), cf. [4, 118, 119, 121], a set of grid points are defined as follows:

\[
g_j^T = \begin{pmatrix} u_1^j & u_2^j & \cdots & u_k^j \end{pmatrix}, \quad j = 1, \cdots, L
\]

\[
u_1^j = \inf_{\gamma \in I_j} \gamma,
\]

\[
u_k^j = \sup_{\gamma \in I_j} \gamma. \quad (2.3)
\]
2.2. Review of the Algorithm of [121]

Figure 2.1: Grids points, parameters and resulting piecewise linear model.

Then by choosing the parameter vectors $\theta_j$ as the values of $f_j(\theta_j, \tilde{u}(t, \omega))$ in the grid points, i.e.

$$
\theta_j = \left( f_{j1}^1 f_{j1}^2 \cdots f_{j1}^k \right)^T, \quad j = 1, \cdots, L
$$

(2.4)

a piecewise linear function of $\tilde{u}(t, \omega)$ can be constructed from the linear segments with end points in $(u_{i-1}^j, f_{i-1}^j)$ and $(u_i^j, f_i^j)$ as shown in Fig. 2.1.

A recursive Gauss-Newton RPEM then follows by a minimization of the cost function

$$
V(\omega, \theta_n) = \lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} E[\varepsilon^2(t, \omega, \theta_n)],
$$

(2.5)

where $\varepsilon(t, \omega, \theta_n)$ is the prediction error defined as

$$
\varepsilon(t, \omega, \theta_n) = y(t) - \tilde{y}(t, \omega, \theta_n),
$$

(2.6)

and $y(t)$ is the measured signal.

In order to formulate the RPEM algorithm, the negative gradient of $\tilde{y}(t, \omega, \theta_n)$ is needed. It is given by (for $\tilde{u}(t, \omega) \in I_j$, $j = 1, \ldots, L$)

$$
\psi(t, \omega, \theta_n) = \left( \frac{\partial f_j(\theta_j, \tilde{u}(t, \omega))}{\partial \omega} \psi(t) \quad 0 \cdots 0 \quad \frac{\partial f_j(\theta_j, \tilde{u}(t, \omega))}{\partial \theta_j} \quad 0 \cdots 0 \right)^T,
$$

(2.7)

where

$$
\psi(t) = t \left. \frac{d\Lambda(\phi)}{d\phi} \right|_{\phi=\omega t}.
$$

(2.8)

The RPEM that appears in [121] was derived as in [69, 119]. It is given by
\[ \varepsilon(t) = y(t) - \hat{y}(t) \]
\[ \lambda(t) = \lambda_0 \lambda(t-1) + 1 - \lambda_0 \]
\[ S(t) = \psi^T(t) P(t-1) \psi(t) + \lambda(t) \]
\[ P(t) = (P(t-1) - P(t-1) \psi(t) S^{-1}(t) \psi^T(t) P(t-1)) / \lambda(t) \]
\[ \left( \hat{\omega}(t) \right) \left( \hat{\theta}_n(t) \right) = \left[ \left( \hat{\omega}(t-1) \right) + \lambda(t) \psi^T(t) \varepsilon(t) \right] D_{\lambda} \]  
\[ (2.9) \]
\[ \hat{u}(t+1) = \Lambda(\phi) \big|_{\phi=\hat{\omega}(t+1)} \]
\[ \psi(t+1) = (t+1) \frac{d\Lambda(\phi)}{d\phi} \big|_{\phi=\hat{\omega}(t+1)} \]

when \( \hat{u}(t+1) \in I_j, j = 1, \ldots, L \)

when \( \hat{u}(t+1) \in [u_{j_i}^i, u_{j+1}^i] \), \( i = 1, \ldots, k_j - 1 \)
\[ \hat{y}(t+1) = \frac{\hat{f}_i(t+1)}{u_{i+1}^j - u_i^j} \frac{u_{i+1}^j - \hat{u}(t+1)}{u_{i+1}^j - u_i^j} \hat{u}(t+1) \]
\[ \frac{\partial f_j}{\partial u} = \frac{\hat{f}_j(t+1) - \hat{f}_j(t)}{u_{i+1}^j - u_i^j} \]
\[ \frac{\partial f_j}{\partial \hat{\omega}} = \frac{u_{i+1}^j - \hat{u}(t+1)}{u_{i+1}^j - u_i^j} \]
\[ \frac{\partial f_j}{\partial \hat{\theta}_j} = 0, \quad l \neq i, i + 1 \]

\[ \frac{\partial f_j}{\partial \theta_j} = \left( \frac{\partial f_j}{\partial f_i^1} \ldots \frac{\partial f_j}{\partial f_i^k_j} \right) \]
\[ \psi(t+1) = \left( \frac{\partial f_j}{\partial u} \psi(t+1) \quad 0 \ldots 0 \quad \frac{\partial f_j}{\partial \theta_j} \right) \]

end

where \( D_{\lambda} \) indicates that the projection algorithms described in [69] are used to keep the estimator in the model set.

It was shown in [121] that the minimum of \( V(\omega, \theta_n) \) is unaffected by colored measurement disturbances. Also, the Cramér-Rao bound (CRB) for \( \langle \omega \theta_n^T \rangle \) was derived in this paper.

### 2.3 Piecewise Quadratic Approximation

The static nonlinearity can also be modeled as a piecewise quadratic function. Since a quadratic curve is uniquely determined by its values in three points,
the following grid points are introduced

\[ g_j = \begin{pmatrix} u_1^j & u_1^j & u_2^j & \cdots & u_{k_j-\frac{1}{2}}^j & u_{k_j}^j \end{pmatrix}^T, \ j = 1, \ldots, L. \]  
(2.10)

The additional grid points are assumed to be located in the middle of the intervals \([u_i^j, u_{i+1}^j], \ i = 1, \ldots, k_j - 1\). The corresponding parameter vector, describing the static nonlinearity is then

\[ \theta_j = \begin{pmatrix} f_1^j & f_1^j & f_2^j & \cdots & f_{k_j-\frac{1}{2}}^j & f_{k_j}^j \end{pmatrix}^T, \ j = 1, \ldots, L. \]  
(2.11)

The RPEM algorithm introduced in (2.9) then requires the following modifications in the nonlinear output and gradient calculations:

when \(\tilde{u}(t+1) \in I_j, \ j = 1, \ldots, L\)

when \(\tilde{u}(t+1) \in [u_i^j, u_{i+1}^j], \ i = 1, \ldots, k_j - 1\)

\[
\begin{align*}
\hat{y}(t+1) &= f_{i+\frac{1}{2}}^j(t) + \frac{\tilde{f}_{i+1}^j(t) - \tilde{f}_i^j(t)}{u_{i+1}^j - u_i^j} (\tilde{u}(t+1) - u_{i+\frac{1}{2}}^j) + \\
&\quad \frac{2(\tilde{f}_{i+1}^j(t) - 2\tilde{f}_{i+\frac{1}{2}}^j(t) + \tilde{f}_i^j(t))}{(u_{i+1}^j - u_i^j)^2} (\tilde{u}(t+1) - u_{i+\frac{1}{2}}^j)^2
\end{align*}
\]

\[
\begin{align*}
\frac{\partial f_j(\cdot)}{\partial \tilde{u}} &= \frac{\tilde{f}_{i+1}^j(t) - \tilde{f}_i^j(t)}{u_{i+1}^j - u_i^j} + \frac{4(\tilde{f}_{i+1}^j(t) - 2\tilde{f}_{i+\frac{1}{2}}^j(t) + \tilde{f}_i^j(t))}{(u_{i+1}^j - u_i^j)^2} (\tilde{u}(t+1) - u_{i+\frac{1}{2}}^j) \\
\frac{\partial f_j(\cdot)}{\partial f_i^j} &= \frac{u_i^j - \tilde{u}(t+1)}{u_{i+1}^j - u_i^j} + \frac{2(\tilde{u}(t+1) - u_{i+\frac{1}{2}}^j)^2}{(u_{i+1}^j - u_i^j)^2} \\
\frac{\partial f_j(\cdot)}{\partial f_{i+\frac{1}{2}}^j} &= 1 - \frac{4(\tilde{u}(t+1) - u_{i+\frac{1}{2}}^j)^2}{(u_{i+1}^j - u_i^j)^2} \\
\frac{\partial f_j(\cdot)}{\partial f_{i+1}^j} &= \frac{\tilde{u}(t+1) - u_{i+\frac{1}{2}}^j}{u_{i+1}^j - u_i^j} + \frac{2(\tilde{u}(t+1) - u_{i+\frac{1}{2}}^j)^2}{(u_{i+1}^j - u_i^j)^2} \\
\frac{\partial f_j(\cdot)}{\partial f_i^j} &= 0, \quad l \neq i, i + \frac{1}{2}, i + 1
\end{align*}
\]

end

\[
\frac{\partial \tilde{f}_j(\cdot)}{\partial \theta_j} = \begin{pmatrix} \frac{\partial f_j(\cdot)}{\partial f_1^j} & \frac{\partial f_j(\cdot)}{\partial f_2^j} & \cdots & \frac{\partial f_j(\cdot)}{\partial f_{k_j-\frac{1}{2}}^j} & \frac{\partial f_j(\cdot)}{\partial f_{k_j}^j} \end{pmatrix}
\]

\[
\hat{\psi}(t+1) = \begin{pmatrix} \frac{\partial f_j(\cdot)}{\partial \tilde{u}} & \frac{\partial \tilde{f}_j(\cdot)}{\partial \psi_i(t+1)} & \cdots & \frac{\partial \tilde{f}_j(\cdot)}{\partial \theta_j} \end{pmatrix}^T
\]

end.

(2.12)
2.4 Numerical Examples

In order to study the performance of the RPEM algorithm suggested for joint estimation of the driving frequency and the parameters of the nonlinear output function, the following simulations were performed.

**Example 2.1.** Convergence to the true parameter vector using Algorithm (2.9).

The data were generated as follows. The driving wave was given by 
\[ u(t, \omega^o) = \sin \omega^o t \]  
where \( \omega^o = 2\pi \times 0.05 \). Two static nonlinearities \((L = 2)\) were used as shown in Fig. 2.2 and defined by

\[ g_j = \begin{pmatrix} -1 & -0.3 & 0 & 0.3 & 1 \end{pmatrix}^T, \ j = 1, 2 \]

\[ \theta^1_o = \begin{pmatrix} -0.8 & -0.3 & 0 & 0.3 & 0.8 \end{pmatrix}^T, \ u \in I_1 \]

\[ \theta^2_o = \begin{pmatrix} -0.8 & -0.5 & 0 & 0.5 & 0.8 \end{pmatrix}^T, \ u \in I_2 \]

where \( u(t, \omega) \in I_1 \) for positive slopes and \( u(t, \omega) \in I_2 \) for negative slopes, respectively. The additive noise was white zero-mean Gaussian with variance \( \sigma^2 = 0.01 \). Algorithm (2.9) was initialized with \( \lambda(0) = 0.95 \), \( \lambda_o = 0.99 \), \( P(0) = 0.01I \) and \( \omega(0) = 2\pi \times 0.02 \). Further, the grid points in (2.13) were used, and the initial values for the nonlinearities were given by straight lines with unity slope.

The modeled signal and the estimated signal are given in Fig. 2.3(a). Also, the estimate of the driving frequency and the parameter estimates are given in Figures 2.3(b)-2.3(d).
2.4. Numerical Examples

Figure 2.3: Convergence using Algorithm (2.9).

After 1500 samples the following estimates were obtained:

\[
\hat{\omega} = 0.3140, \\
\hat{\theta}_1 = \begin{pmatrix} -0.784 & -0.284 & -0.037 & 0.326 & 0.796 \end{pmatrix}^T, \\
\hat{\theta}_2 = \begin{pmatrix} -0.802 & -0.515 & -0.085 & 0.486 & 0.796 \end{pmatrix}^T.
\]

Hence, it can be concluded that convergence to the true parameter vector is taking place.

**Example 2.2.** *Convergence to the true parameter vector using Algorithm (2.12).*

In this example the data were generated and Algorithm (2.12) was initialized
as in Example 2.1. The grid points in this case were
\[ g_j = ( -1, -0.5, 0, 0.5, 1 )^T, \quad j = 1, 2 \]
\[ \theta^*_1 = ( -0.8, -0.5, 0, 0.5, 0.8 )^T, \quad u \in I_1 \]
\[ \theta^*_2 = ( -0.8, -0.6, 0, 0.6, 0.8 )^T, \quad u \in I_2. \]

The estimated fundamental frequency is given in Fig. 2.4. After 1500 samples the following estimates were obtained:
\[ \hat{\omega} = 0.3143, \]
\[ \hat{\theta}_1 = ( -0.804, -0.536, -0.069, 0.470, 0.796 )^T, \]
\[ \hat{\theta}_2 = ( -0.835, -0.575, 0.066, 0.613, 0.823 )^T. \]

Also here it can be concluded that the convergence to the true parameter vector is taking place.

**Example 2.3. Tracking fundamental frequency variations.**

To improve the ability of Algorithm (2.9) to track fundamental frequency variations, it can be modified as follows.
\[
\varepsilon(t) = y(t) - \hat{y}(t) \\
S(t) = \psi^T(t)P(t-1)\psi(t) + r_2(t) \\
P(t) = P(t-1) - P(t-1)\psi(t)S^{-1}(t)\psi^T(t)P(t-1) + R_1(t) \\
\begin{pmatrix}
\hat{\omega}(t) \\
\hat{\theta}_n(t)
\end{pmatrix}
= \begin{pmatrix}
\hat{\omega}(t-1) \\
\hat{\theta}_n(t-1)
\end{pmatrix}
+ P(t)\psi(t)\varepsilon(t) \quad (2.15)
\]

where \( R_1(t) \) and \( r_2(t) \) are the gain design variables (see [69] page 273). This modification transforms the problem into an extended Kalman filter (EKF) formulation. For more literature on tracking, see [34, 42, 43, 115].
Figure 2.5: Tracking the fundamental frequency using Algorithm (2.15).

The data were generated as in Example 2.1 and Algorithm (2.15) was initialized with $P(0) = 0.01I$ and $\omega(0) = 2\pi \times 0.02$. The design variables were $R_1(t) = 10^{-4}I$ and $r_2(t) = 0.2$. Also, the grid points in (2.13) were used, and the initial values for the nonlinearities were given by straight lines with unity slope.

The true and the estimated fundamental frequency are given in Fig. 2.5. After 2400 samples the following estimates were obtained:

$\hat{\omega} = 0.3479,$

$\hat{\theta}_1 = \begin{pmatrix} -0.817 & -0.259 & 0.020 & 0.342 & 0.800 \end{pmatrix}^T,$

$\hat{\theta}_2 = \begin{pmatrix} -0.799 & -0.487 & -0.091 & 0.378 & 0.839 \end{pmatrix}^T,$

and it can be concluded that Algorithm (2.15) has the ability to track the fundamental frequency variations.

**Example 2.4. Statistical results for the fundamental frequency estimate.**

In order to compare the performance of the RPEM algorithm given in (2.9) with the derived Cramér-Rao bound (CRB) in [121] for the fundamental frequency estimation, 100 Monte-Carlo simulation experiments were performed.

The data were generated and the algorithm was initialized as in Example 2.1 with $\omega(0) = 2\pi \times 0.03$. The statistics is based on excluding simulations that did not satisfy a margin of 5 standard deviations (as predicted by the CRB) from the true fundamental frequency. Both the CRB of the fundamental frequency estimate and the mean square error (MSE) value were evaluated for different signal to noise ratios (SNRs). The data length was 2000 samples. The number of excluded simulations did not exceed 15 experiments. The statistical results are plotted in Fig. 2.6. The results show that the RPEM algorithm gives an accurate estimate of the fundamental frequency at moderate and high SNRs.
Figure 2.6: The CRB and the MSE of the fundamental frequency estimate using Algorithm (2.9).

2.5 Conclusions

The recursive periodic signal estimation approach introduced in [121] has been studied by numerical examples to investigate the convergence to the true parameter vector and the ability of tracking fundamental frequency variations. Modifications that improve the latter property were suggested. Also, the approximation of the static nonlinearity by a piecewise quadratic parameterization was introduced. Moreover, Monte-Carlo experiments show that the method gives good results, in particular for moderate values of the signal to noise ratio.
Chapter 3

A Modified Nonlinear Approach to Periodic Signal Modeling

3.1 Introduction

The periodic signal modeling technique introduced in Chapter 2 is modified in this chapter. The difference between the technique described in Chapter 2 and the one given here is that in [121] the differential static gain was fixed in the linear block. Here, however, the differential static gain is fixed in the nonlinear block. This requires a slight modification of the parameterization of the static nonlinearity.

The modifications in the proposed RPEM are obtained by introducing an interval in the nonlinear block with fixed gain. The modification in the convergence analysis is, however, substantial and allows a treatment of the local convergence properties of the algorithm. This is the main reason for the modification. The convergence analysis is based on Ljung’s method with an associated ordinary differential equation (ODE), see [65, 66, 69].

The contributions of this chapter can hence be summarized as follows. Conditions for local convergence to the true parameter vector are derived with averaging theory for the suggested method. Furthermore, the Cramér-Rao bound (CRB) is calculated. Finally, the performance for joint estimation of the driving frequency and the parameters of the nonlinear output function is studied by numerical examples.

The chapter is organized as follows. Modifications in the algorithm of [121] are discussed in Section 3.2. A local convergence analysis is presented in Section 3.3. Section 3.4 presents the derivation of the CRB for the modified algorithm. In Section 3.5, some numerical examples are given. Conclusions are given in Section 3.6.

3.2 The Modified Algorithm

Since the model is a cascade of two blocks, the differential static gain of the model will be a factor of two free parameters. It is nevertheless necessary for the algorithm to have information about where the static gain is located, or the criterion function may have an infinite number of minima. In this case,
the RPEM algorithm will be sliding around in a valley of the criterion function resulting in numerical problems. Therefore, one of the parameters need to be fixed, cf. [118]. In [121], this was done in the driving signal block. Here, however, the opposite situation will be investigated.

In order to fix the static gain in an amplitude subinterval \( I_o \) contained in exactly one of the \( L \) disjoint intervals \( I_j, j = 1, \cdots, L \); the driving input signal of the modified method is modeled as

\[
\hat{u}(t, \theta_i) = X \Lambda(\omega t),
\]

\[
\theta_i = (X \omega)^T,
\]  

(3.1)

where \( X \) is a (possibly time varying) parameter recursively estimated to allow the linear block of the model to adapt its static gain so that the data in \( I_o \) can be explained. Choosing \( I_o \) to be contained in the first interval \( I_1 \), \( f_1(\theta_1, \hat{u}(t, \theta_i)) \) is defined as in [118] to become

\[
f_1(f_o, \theta_1, \hat{u}(t, \theta_i)) = K_o \hat{u}(t, \theta_i) + f_o, \quad \hat{u}(t, \theta_i) \in I_o \subset I_1, \tag{3.2}
\]

so

\[
\frac{\partial f_1(f_o, \theta_1, \hat{u}(t, \theta_i))}{\partial \hat{u}} = K_o, \quad \hat{u}(t, \theta_i) \in I_o. \tag{3.3}
\]

Here \( K_o \) is a user chosen constant. The grid points now become [118]

\[
\varrho_j^T = \left\{ \begin{array}{c}
(u_{1-k_j} \cdots u_{1-k_j+1} \cdots u_{o-} \cdots u_{o+} \cdots u_{k_j+1} \cdots u_{k_j-1}; \ j = 1 \\
(u_{-k_j} \cdots u_{k_j-1} \cdots u_{-1} \cdots u_{j-1} \cdots u_{k_j+1} \cdots u_{k_j-1}; \ j = 2, \cdots, L.
\end{array} \right.
\]  

(3.4)

Hence, the parameters \( \theta_j \) are defined by

\[
\theta_j = \left( f_{k_j}^j \cdots f_{-1}^j f_1^j \cdots f_{k_j}^j \right)^T, \ j = 1, \cdots, L
\]

(3.5)

\[
f_j(\theta_j, u_i^1) = \left\{ \begin{array}{l}
K_o \hat{u}(t, \theta_i) + f_o, \quad \hat{u}(t, \theta_i) \in I_o, \ j = 1 \\
f_i^1, \quad i = -k_j, \cdots, -1, 1, \cdots, k_j, \ j = 1, \cdots, L.
\end{array} \right.
\]

Note that there are no parameters corresponding to the grids \( u_{o-} \) and \( u_{o+} \), since

\[
f_1(f_o, \theta_1, u_{o-}) = K_o u_{o-} + f_o,
\]

\[
f_1(f_o, \theta_1, u_{o+}) = K_o u_{o+} + f_o.
\]

(3.6)

The parameter vector then takes the form, with \( \theta_i \) defined by (3.1)

\[
\theta = (\theta_1^T \cdots \theta_n^T)^T,
\]

\[
\theta_n = (f_o \cdots f_n^T)^T,
\]

(3.7)

\[
\tilde{\theta}_n = (\theta_1^T \cdots \theta_n^T)^T,
\]

\[
\theta_j = (f_{-k_j}^j \cdots f_{-1}^j f_1^j \cdots f_{k_j}^j)^T, \ j = 1, \cdots, L.
\]
Taking into account that \( \psi_1(t) \) in this case is given by

\[
\psi_1(t) = \left( \Lambda(\phi) \big|_{\phi=\omega t} \right. X \left. \frac{d \Lambda(\phi)}{d \phi} \big|_{\phi=\omega t} \right)^T,
\]

(3.8)

the recursive prediction error algorithm of Chapter 2 is modified to become

\[
\begin{align*}
\varepsilon(t) &= y(t) - \hat{y}(t) \\
\lambda(t) &= \lambda_o \lambda(t - 1) + 1 - \lambda_o \\
S(t) &= \psi^T(t) \mathbf{P}(t - 1) \hat{\psi}(t) + \lambda(t) \\
\mathbf{P}(t) &= \left( \mathbf{P}(t - 1) - \mathbf{P}(t - 1) \hat{\psi}(t) S^{-1}(t) \psi^T(t) \mathbf{P}(t - 1) / \lambda(t) \right) \\
\left( \begin{array}{c} \hat{\theta}_1(t) \\ \hat{\theta}_n(t) \end{array} \right) &= \left[ \left( \begin{array}{c} \hat{\theta}_1(t - 1) \\ \hat{\theta}_n(t - 1) \end{array} \right) + \mathbf{P}(t) \hat{\psi}(t) \varepsilon(t) \right] \text{ } D_M \\
\hat{u}(t + 1) &= \hat{X}(t) \Lambda(\phi) \big|_{\phi=\omega(t)(t+1)} \\
\psi_1(t + 1) &= \left( \Lambda(\phi) \big|_{\phi=\omega(t)(t+1)} \right. \hat{X}(t)(t + 1) \left. \frac{d \Lambda(\phi)}{d \phi} \big|_{\phi=\omega(t)(t+1)} \right)^T
\end{align*}
\]

when \( \hat{u}(t + 1) \in I_1 \)

\[
\hat{y}(t + 1) = \tilde{f}_1(t + 1) u_{o-} - \left( \frac{K_o u_{o-} + \tilde{f}_o(t)}{u_{o-} - u_{l-1}} \right) u_{o-} - \left( \frac{K_o u_{o-} + \tilde{f}_o(t) - \tilde{f}_{l-1}t}{u_{o-} - u_{l-1}} \right) \hat{u}(t + 1)
\]

\[
\begin{align*}
\frac{\partial f_1(\cdot)}{\partial u} &= \frac{K_o u_{o-} + \tilde{f}_o(t) - \tilde{f}_{l-1}t}{u_{o-} - u_{l-1}} \\
\frac{\partial f_1(\cdot)}{\partial f_0} &= \frac{u_{o-} - \tilde{u}(t + 1)}{u_{o-} - u_{l-1}} \\
\frac{\partial f_1(\cdot)}{\partial f_l} &= \frac{\tilde{u}(t + 1) - u_{l-1}}{u_{o-} - u_{l-1}} \\
\frac{\partial f_1(\cdot)}{\partial f_l^l} &= 0, \quad l \neq -1, 0
\end{align*}
\]

end

when \( \hat{u}(t + 1) \in [u_{o-}, u_{o+}] \)

\[
\hat{y}(t + 1) = K_o \hat{u}(t + 1) + \tilde{f}_o(t)
\]

\[
\begin{align*}
\frac{\partial f_1(\cdot)}{\partial u} &= K_o \\
\frac{\partial f_1(\cdot)}{\partial f_0} &= 1 \\
\frac{\partial f_1(\cdot)}{\partial f_l^l} &= 0, \quad l \neq 0
\end{align*}
\]

end
when \( \hat{u}(t+1) \in [u_{o+}, u_1^j] \)

\[
\hat{y}(t+1) = \left( K_o u_{o+} + \hat{f}_o(t) \right) u_1^j - \hat{f}_1^j(t) u_{o+} + \frac{\hat{f}_1^j(t) - K_o u_{o+} - \hat{f}_o(t)}{u_1^j - u_{o+}} \hat{u}(t+1)
\]

\[
\frac{\partial f_j(\cdot)}{\partial \hat{u}} = \frac{\hat{f}_1^j(t) - K_o u_{o+} - \hat{f}_o(t)}{u_1^j - u_{o+}}
\]

\[
\frac{\partial f_j(\cdot)}{\partial \hat{f}_0} = \frac{u_1^j - \hat{u}(t+1)}{u_1^j - u_{o+}}
\]

\[
\frac{\partial f_j(\cdot)}{\partial \hat{f}_j^l} = \frac{\hat{u}(t+1) - u_1^j}{u_{l+1}^j - u_1^j}
\]

\[
\frac{\partial f_j(\cdot)}{\partial \hat{f}_+^j} = \frac{u_{l+1}^j - \hat{u}(t+1)}{u_{l+1}^j - u_1^j}
\]

\[
\frac{\partial f_j(\cdot)}{\partial \hat{f}_-^j} = 0, \quad l \neq i, i+1
\]

\[
\frac{\partial f_j(\cdot)}{\partial \hat{f}_j^l} = 0, \quad l \neq 0, 1
\]

end

when \( \hat{u}(t+1) \in I_j, \quad j = 1, \ldots, L \)

\[
\hat{y}(t+1) = \frac{\hat{f}_1^j(t) u_{l+1}^j - \hat{f}_+^j(t) u_1^j}{u_{l+1}^j - u_1^j} + \frac{\hat{f}_+^j(t) - \hat{f}_1^j(t)}{u_{l+1}^j - u_1^j} \hat{u}(t+1)
\]

\[
\frac{\partial f_j(\cdot)}{\partial \hat{u}} = \frac{\hat{f}_+^j(t) - \hat{f}_1^j(t)}{u_{l+1}^j - u_1^j}
\]

\[
\frac{\partial f_j(\cdot)}{\partial \hat{f}_0} = \frac{u_{l+1}^j - \hat{u}(t+1)}{u_{l+1}^j - u_1^j}
\]

\[
\frac{\partial f_j(\cdot)}{\partial \hat{f}_j^l} = \frac{\hat{u}(t+1) - u_1^j}{u_{l+1}^j - u_1^j}
\]

\[
\frac{\partial f_j(\cdot)}{\partial \hat{f}_+^j} = \frac{u_{l+1}^j - \hat{u}(t+1)}{u_{l+1}^j - u_1^j}
\]

\[
\frac{\partial f_j(\cdot)}{\partial \hat{f}_-^j} = 0
\]

\[
\frac{\partial f_j(\cdot)}{\partial \hat{f}_j^l} = 0, \quad l \neq i, i+1
\]

end

\[
\frac{\partial f_j(\cdot)}{\partial \theta_i} = \left( \frac{\partial f_j(\cdot)}{\partial \theta_{k_j^-}} \ldots \frac{\partial f_j(\cdot)}{\partial \theta_{k_j^+}} \right)
\]

\[
\psi(t+1) = \left( \frac{\partial f_j(\cdot)}{\partial \theta_k} \psi(t+1) \frac{\partial f_j(\cdot)}{\partial \theta_k} 0 \ldots 0 \frac{\partial f_j(\cdot)}{\partial \theta_k} 0 \ldots 0 \right)^T
\]

end.

In this chapter the nonlinearity is chosen to be piecewise linear with the estimated parameters being the function values in a fixed set of grid points resulting in a fixed grid point adaptation. These grid points are chosen by the user depending on any prior information about the wave form. For example the peak to peak value of the wave form helps to determine the grids range for each in-
3.3 Local Convergence

As mentioned above, one major reason for the modification of the original method of [121], was to perform a convergence analysis. Local convergence of the RPEM to the true parameter vector is analyzed here with the linearized, associated ordinary differential equation (ODE), see e.g. [69, 118]. First, a brief introduction to the associated ODE approach is given. Second, the details of the local convergence analysis are presented. Finally, a discussion on the technical conditions needed to guarantee local convergence is given.

3.3.1 The Associated ODE Approach

The associated ODE approach was used in [69] to perform the convergence analysis of a basic structure of recursive algorithms related to quadratic criteria such as (2.5). This section reviews the relevant parts of the presentation of [69]. The basic structure of [69] is

\[
\varepsilon(t) = y(t) - \hat{y}(t)
\]

\[
R(t) = R(t-1) + \gamma(t)\left[\eta(t)\Lambda^{-1}(t)\eta^T(t) - R(t-1)\right]
\]

\[
\hat{\theta}(t) = \hat{\theta}(t-1) + \gamma(t)R^{-1}(t)\eta(t)\Lambda^{-1}(t)\varepsilon(t)
\]

\[
\xi(t+1) = \left(\begin{array}{c} \hat{y}(t+1) \\ \text{col } \eta(t+1) \end{array}\right)
\]

\[
= C(\hat{\theta}(t))\xi(t+1)
\]

where \( R(t) \) is the Hessian approximation in the Gauss-Newton algorithm, \( \gamma(t) \) is the adaptation gain (\( \gamma(t) \rightarrow 0 \) as \( t \rightarrow \infty \)), \( \eta(t) \) is a vector that is related to the gradient of the prediction \( \hat{y}(t) \) w.r.t. \( \hat{\theta} \), \( \Lambda(t) \) is the covariance matrix of prediction errors, \( \{z(t)\} \) is the data set, \( A(\hat{\theta}(t)) \), \( B(\hat{\theta}(t)) \) and \( C(\hat{\theta}(t)) \) are extended state space matrices that result from merging two finite-dimensional linear filters which produce the output \( \hat{y}(t) \) and the gradients \( \eta(t) \). \( \xi(t) \) denotes the states of this extended state space model.

Now, the performance of the recursive algorithm (3.10)-(3.14) is investigated for sufficiently large \( t \). This means that \( \gamma(t) \) will be small, the estimates \( \{\hat{\theta}(t)\} \) will change slowly and closely to some value \( \bar{\theta} \), and \( R(t) \) will be close to some value \( \bar{R} \). In this case, Eqs. (3.11)-(3.12) can be modified to

\[
\hat{\theta}(t) \approx \hat{\theta}(t-1) + \gamma(t)\Lambda^{-1}(t)\varepsilon(t, \hat{\theta}),
\]

\[
R(t) \approx R(t-1) + \gamma(t)\left[\eta(t, \hat{\theta})\Lambda^{-1}(t, \hat{\theta})\varepsilon(t, \hat{\theta}) - \bar{R}\right].
\]
Introducing the following average updating directions:

\[ f(\bar{\theta}) \triangleq E[\eta(t, \bar{\theta})\Lambda^{-1}\varepsilon(t, \bar{\theta})], \]  

\[ G(\bar{\theta}) \triangleq E[\eta(t, \bar{\theta})\Lambda^{-1}\eta^T(t, \bar{\theta})], \]  

equations (3.15)-(3.16) can be written in the form

\[ \hat{\theta}(t) \approx \hat{\theta}(t - 1) + \gamma(t)\bar{R}^{-1}f(\bar{\theta}) + \gamma(t)v_1(t), \]  

\[ R(t) \approx R(t - 1) + \gamma(t)[G(\bar{\theta}) - \bar{R}] + \gamma(t)v_2(t), \]  

where \( v_1(t) \) and \( v_2(t) \) are zero-mean random variables.

The analysis of [69] proceeds by defining times \( t \) and \( \bar{t} \) such that

\[ \sum_{k=t}^{\bar{t}} \gamma(k) = \Delta \tau \]  

where \( \Delta \tau \) is a small number. This follows since \( t \) was assumed to be sufficiently large and \( \gamma(t) \) is arbitrarily small. Hence, if \( \hat{\theta}(t) = \bar{\theta} \) and \( R(t) = \bar{R} \), Eqs. (3.19)-(3.20) can be written in the form

\[ \hat{\theta}(\bar{t}) \approx \bar{\theta} + \Delta \tau \bar{R}^{-1}f(\bar{\theta}) + \sum_{k=t}^{\bar{t}} \gamma(k)v_1(k) \]  

\[ R(\bar{t}) \approx \bar{R} + \Delta \tau [G(\bar{\theta}) - \bar{R}] + \sum_{k=t}^{\bar{t}} \gamma(k)v_2(k) \]  

since \( \sum_{k=t}^{\bar{t}} \gamma(k)v_1(k) \ll \Delta \tau \bar{R}^{-1}f(\bar{\theta}) \) and \( \sum_{k=t}^{\bar{t}} \gamma(k)v_2(k) \ll \Delta \tau [G(\bar{\theta}) - \bar{R}] \).

Now, by changing the time scale such that \( t \leftrightarrow \tau \) and \( \bar{t} \leftrightarrow \tau + \Delta \tau \), Eqs. (3.22)-(3.23) can be regarded as a numerical scheme to solve the following ODE:

\[ \dot{\theta} = R^{-1}f(\theta) \]  

\[ \dot{R} = G(\theta) - R \]  

where \( \theta \) is a fixed vector belonging to \( D_M \), the set of parameter vectors describing the model set.

Finally, the analysis of [69] linked the stability properties of the solution

\[ x = \begin{pmatrix} \theta \\ \text{col} \ R \end{pmatrix} \]  

of the associated ODE (3.24) to the convergence of the recursive algorithm (3.10)-(3.14). In brief, if the ODE is globally or locally stable, the recursive algorithm will be globally or locally convergent, respectively.
3.3.2 Analysis Details

The convergence analysis of the RPEM algorithm (3.9) proceeds along the following lines. First, it is investigated when the true parameter vector is a stationary point to the differential equation associated with (3.9). The local stability of this point is then studied. The analysis relies on the fact that Ljung’s original method of analysis is also applicable to the Wiener model structure. This was shown formally in [119]. However, here the driving linear block is slightly different from [119].

The celebrated analysis of [65] relies on the assumption that the parameters of the model should be constrained to the set $D_M$, where strict asymptotic stability holds. This cannot be achieved in the present case, since the periodic driving function is only BIBO-stable (when modeled as a filter). Therefore, a completely stringent treatment of the local convergence properties of the present algorithm, requires modifications of the approach of [65]. This will not be undertaken here, since the periodic function can be treated as a superposition of sine waves. Since any sine wave can be represented with an AR-model with separate poles on the unit-circle, the present case can thus be thought as a limiting case of [65]. Because of the simplicity of the model (the static approach), this heuristic argument is believed to be sufficient to motivate the following calculations.

The local convergence analysis of Algorithm (3.9) begins by calculating the average updating directions (3.17)-(3.18) that define the associated ODE from the model and gradient relations. The calculations consider using a fixed parameter vector $\theta \in D_M$ and a fixed $R$, as mentioned in Section 3.3.1. When Algorithm (3.9) is compared to the algorithm of [118], it is found that the average updating directions are

$$f(\theta) = \lim_{t \to \infty} E \psi(t, \theta) \varepsilon(t, \theta)$$

$$= \lim_{t \to \infty} E \left( \begin{array}{c} \frac{\partial f_j(\cdot)}{\partial \theta_n} \psi_j(t, \theta) \varepsilon(t, \theta) \\ \frac{\partial f_j(\cdot)}{\partial \theta_o} \varepsilon(t, \theta) \end{array} \right),$$

(3.26)

$$F(R, \theta) = G(\theta) - R,$$

(3.27)

$$G(\theta) = \lim_{t \to \infty} E \psi(t, \theta) \psi^T(t, \theta)$$

$$= \lim_{t \to \infty} E \left( \begin{array}{c} \frac{\partial f_j(\cdot)}{\partial \theta_n} \psi_j \psi^T_j \\ \frac{\partial f_j(\cdot)}{\partial \theta_o} \psi_j \psi^T_o \\ \frac{\partial f_j(\cdot)}{\partial \theta_m} \psi_j \psi^T_m \\ \frac{\partial f_j(\cdot)}{\partial \theta_n} \frac{\partial f_j(\cdot)}{\partial \theta_o} \psi_j \psi^T_o \\ \frac{\partial f_j(\cdot)}{\partial \theta_o} \frac{\partial f_j(\cdot)}{\partial \theta_m} \psi_j \psi^T_m \\ \frac{\partial f_j(\cdot)}{\partial \theta_m} \frac{\partial f_j(\cdot)}{\partial \theta_o} \psi_j \psi^T_o \\ \frac{\partial f_j(\cdot)}{\partial \theta_m} \frac{\partial f_j(\cdot)}{\partial \theta_m} \psi_j \psi^T_m \\ \frac{\partial f_j(\cdot)}{\partial \theta_n} \frac{\partial f_j(\cdot)}{\partial \theta_o} \frac{\partial f_j(\cdot)}{\partial \theta_m} \psi_j \psi^T_o \end{array} \right).$$

(3.28)

Proceeding with the analysis, the following condition is introduced;

**Assumption 3.1.** The linear block and the static nonlinearity of the system are contained in the model set.

Noticing that the model output is given by

$$\hat{y}(t, \theta) = f_j(\theta_n, \tilde{u}(t, \theta_1)), \quad \tilde{u}(t, \theta_1) \in I_j, \ j = 1, \ldots, L$$

(3.29)
then there is a vector $\theta^o$ such that the output of the static nonlinearity of the system is described by

$$y(t) = f_j(\theta^o, \tilde{u}(t, \theta^o)) + w(t), \quad \tilde{u}(t, \theta^o) \in I_j, \quad j = 1, \ldots, L. \quad (3.30)$$

where $w(t)$ is the disturbance which satisfies the following condition:

(c.f. [69, 118])

**Assumption 3.2.** $w(t)$ is a bounded, strictly stationary, zero mean stochastic process that fulfills

$$E |w(t) - w^o(t)|^4 \leq c \lambda^{t-s}, \quad c < \infty, \quad |\lambda| < 1.$$  

**Remark 3.1.** $w^o(t)$ denotes a random variable that belongs to the $\sigma$-algebra generated by $\{w(t)\}, \forall t \geq s$ but independent of $\{w(t)\}^t_s$. This is simply means that $w^o(t)$ is a very good approximation of $w(t)$ during the time interval $[s, t]$. See [69] for more details.

Since there is no use of $w(t)$ in the input signal generation, the following condition is also satisfied;

**Assumption 3.3.** $\psi_l(t, \theta^o)$ and $w(t)$ are independent.

By choosing $\theta = \theta^o$, it is concluded from (3.9) that

$$\varepsilon(t, \theta^o) = y(t) - \hat{y}(t, \theta^o) = w(t). \quad (3.31)$$

When this is inserted into (3.26) the result $f(\theta^o) = 0$ is obtained by Assumption 3.3. Assumption 3.2, together with (3.27) then implies that the ODE (3.24) associated with (3.9) has a stationary point described by

$$x^* \triangleq \left( \begin{array}{c} \theta^o \\ \text{col} G(\theta^o) \end{array} \right). \quad (3.32)$$

Following [69, 118], the local stability of the stationary point (3.32) need to be checked. This can be done by studying the eigenvalues of the matrix

$$A \triangleq \left. \frac{\partial}{\partial x} \left( \begin{array}{c} R^{-1} f(\theta) \\ \text{col}(G(\theta) - R) \end{array} \right) \right|_{x=x^*} = \left( \begin{array}{cc} G^{-1}(\theta^o) \frac{df(\theta)}{d\theta}|_{\theta=\theta^o} & 0 \\ \frac{df(\theta)}{d\theta}|_{\theta=\theta^o} & 0 \end{array} \right). \quad (3.33)$$

Hence, $x^*$ is locally stable provided that all the eigenvalues of $A$ are in the LHP. This means that it is necessary to prove that the eigenvalues of $G^{-1}(\theta^o) \frac{df(\theta)}{d\theta}|_{\theta=\theta^o}$ are in the LHP. The derivative of $f(\theta)$ w.r.t. $\theta$ can be evaluated from (3.26) by straightforward calculations. When $\theta$ is replaced by $\theta^o$ in the resulting expression, Eq. (3.28) together with Assumption 3.3 gives

$$G^{-1}(\theta^o) \frac{df(\theta)}{d\theta}|_{\theta=\theta^o} = -I. \quad (3.34)$$
Equation (3.34) is valid provided that the inverse of the matrix $G(\theta^o)$ exists. Taking into account that $G(\theta^o)$ is at least positive semidefinite by construction, the goal of the next analysis is to find conditions that imply the positive definiteness of $G(\theta^o)$.

In order to achieve the previous goal, it is convenient, as done in [117], to study the contribution to $G(\theta^o)$ from the mid-subinterval $I_o$. Introducing the gate function for this purpose, where

$$\text{gate}(\hat{u}(t, \theta_i^o)) = \begin{cases} 1, & \hat{u}(t, \theta_i^o) \in I_o \subset I_1 \\ 0, & \text{otherwise.} \end{cases} \quad (3.35)$$

Again, by $\hat{u}(t, \theta_i^o) \in I_o$ it is meant that the phase $\omega t$ is such that $I_o$ is in effect. Thus $G(\theta^o)$ can then be expressed as

$$G(\theta^o) = \lim_{t \to \infty} E (1 - \text{gate}(\hat{u})) \psi(t, \theta) \psi^T(t, \theta) + \lim_{t \to \infty} E \text{gate}(\hat{u}) \psi(t, \theta) \psi^T(t, \theta),$$

$$= \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} + \begin{pmatrix} \tilde{A} & 0 \\ 0 & 0 \end{pmatrix}, \quad (3.36)$$

where (cf. Eq. (3.28))

$$\tilde{A} = \lim_{t \to \infty} E \text{gate}(\hat{u}) \begin{pmatrix} K_o^2 \psi_1 \psi_1^T & K_o \psi_1^T \\ K_o \psi_1 & 1 \end{pmatrix}_{|\theta_i = \theta_i^o}, \quad (3.37)$$

$$C = \lim_{t \to \infty} E (1 - \text{gate}(\hat{u})) \frac{\partial f_j(\cdot)}{\partial \theta_n} \frac{\partial f_j(\cdot)}{\partial \theta_n} \bigg| \theta_n = \tilde{\theta}_n^o$$

$$= \lim_{t \to \infty} E (1 - \text{gate}(\hat{u})) \begin{pmatrix} \frac{\partial f_1(\cdot)}{\partial \theta_1} & \frac{\partial f_L(\cdot)}{\partial \theta_1} \\ \vdots & \vdots \\ \frac{\partial f_1(\cdot)}{\partial \theta_L} & \frac{\partial f_L(\cdot)}{\partial \theta_L} \end{pmatrix} \bigg| \theta_n = \tilde{\theta}_n^o \quad (3.38)$$

Now Lemma 2 of [118] will be useful. This result can be formulated as follows.

**Lemma 3.1.** Consider the block-matrix decomposition

$$G = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} + \begin{pmatrix} \tilde{A} & 0 \\ 0 & 0 \end{pmatrix},$$

where both terms on the right-hand side are symmetric. Assume that the first term of $G$ is positive semidefinite and that $\tilde{A}$ and $C$ are positive definite. Then $G$ is also positive definite.

**Proof.** See [118].

The following two lemmas give the conditions necessary to satisfy that $\tilde{A}$ and $C$ are positive definite. The notation intentionally follows [117, 118].
Lemma 3.2. Assume that the following conditions hold for the RPEM:

Assumption 3.4. \( \alpha_o = \lim_{t \to \infty} E \text{gate}(\hat{u}) > 0 \).

Assumption 3.5. \( \lim_{t \to \infty} E \text{gate}(\hat{u}) \Delta \psi_l \Delta \psi_l^T \geq \delta \lim_{t \to \infty} E \text{gate}(\hat{u}) \psi_l^o \psi_l^o + \Delta \psi_l \Delta \psi_l^T \), \( \delta > 0 \), where \( \psi_l = \frac{1}{\alpha_o} \psi_l^o + \Delta \psi_l \) and \( \psi_l^o = \lim_{t \to \infty} E \text{gate}(\hat{u}) \psi_l \).

Assumption 3.6. \( |K_o| > 0 \).

Assumption 3.7. \( \psi_l \) is unbiased, \( \text{i.e.} \lim_{t \to \infty} E \psi_l = 0 \).

Assumption 3.8. \( \Lambda(\phi) \) is continuous, three times differentiable where \( |\Lambda(\phi_o)| \geq \epsilon_1 > 0 \), \( \frac{d^3 \Lambda(\phi)}{d\phi^3}\big|_{\phi=\phi_o} \geq \epsilon_2 > 0 \), and has at least one minimum or one maximum point at \( \phi_o \).

Assumption 3.9. \( \phi = \omega t \) has a continuous probability density function (pdf).

Then \( \hat{A} > 0 \).

Proof. See Appendix 3.A.

Lemma 3.3. Assume that the following condition is satisfied for the model;

Assumption 3.10. The probability density function \( h_{\hat{u}}(\hat{u}) \) of \( \hat{u}(t, \theta_o^j) \) fulfills \( h_{\hat{u}}(\hat{u}) \geq \delta_l > 0 \) in at least one nonzero interval \( [a^l_i, b^l_i] \subseteq I^l_i = (u^l_{i-1}, u^l_i) \) for all \( i = -k^-_j + 1, \ldots, k^+_j, \ j = 1, \ldots, L \).

Then \( C > 0 \).

Proof. See Appendix 3.B.

The fact that \( \hat{A} \) and \( C \) are positive definite now implies the positive definiteness of \( G(\theta^o) \) using Lemma 3.1. This completes the proof of the following theorem:

Theorem 3.1. Under Assumptions 2.1-2.2 and Assumptions 3.1-3.10, the RPEM algorithm given in (3.9) converges locally to

\[
\left( \begin{array}{c} \theta^c_j \\ \theta^o_n \end{array} \right) \in \mathcal{D}_M.
\]

3.3.3 Discussion

The conditions of Lemma 3.2 and Lemma 3.3, except Assumptions 3.7-3.9, are similar to the corresponding ones of [118]. Assumptions 3.4-3.6 are necessary to indicate that there is a sufficient amount of signal energy in the mid-subinterval \( I_o \). Assumption 3.10 means that there should be signal energy somewhere in every subinterval of the piecewise linear model of the static nonlinearity.
The analysis of Appendix 3.A leads to the inequality
\[
\lim_{t \to \infty} E \Delta \tilde{\psi}_t \Delta \tilde{\psi}_t^T > 0, \tag{3.40}
\]
where \(\Delta \psi_t = \Delta \psi_t^o + \Delta \tilde{\psi}_t\) and \(\Delta \psi_t^o = \lim_{t \to \infty} E \Delta \psi_t\), cf. Appendix 3.A. The inequality (3.40) can not be satisfied for all driving signals as shown in the following two examples.

Example 3.1. Consider the following
\[
\hat{u}(t, \theta^o_t) = X^o(t) \Lambda(\phi)|_{\phi = \omega_o t},
\]
\[
\Lambda(\phi) = \sin(\phi), \tag{3.41}
\]
and assume that the phase \(\omega_o t\) is uniformly distributed in the interval \([0, 2\pi]\). In this case \(\Delta \tilde{\psi}_t = \psi_t\), because there is no bias in \(\psi_t\). Thus
\[
E \Delta \tilde{\psi}_t \Delta \tilde{\psi}_t^T = E \left( \frac{X^o}{\omega} \phi \Lambda(\phi) \frac{\partial \Lambda(\phi)}{\partial \phi} \left( \frac{X^o}{\omega^2} \right) \phi^2 \left[ \frac{\partial \Lambda(\phi)}{\partial \phi} \right]^2 \right)_{\phi = \omega_o t}. \tag{3.42}
\]
Straightforward calculations give
\[
E \Delta \tilde{\psi}_t \Delta \tilde{\psi}_t^T = \frac{1}{2\pi} \int_0^{2\pi} \Delta \tilde{\psi}_t \Delta \tilde{\psi}_t^T d\phi
\]
\[
= \left( \frac{1}{\frac{X^o}{\omega}} \left[ \frac{X^o}{\omega^2} \right]^2 \left( \frac{2\pi^2}{\frac{X^o}{\omega}} + \frac{1}{4} \right) \right), \tag{3.43}
\]
which is a positive definite matrix. Thus (3.40) is a valid inequality in this case.

Example 3.2. Consider the following
\[
\hat{u}(t, \theta^o_t) = X^o(t) \Lambda(\phi)|_{\phi = \omega_o t},
\]
\[
\Lambda(\phi) = \frac{1}{2\pi} (\phi - \pi), \tag{3.44}
\]
\[
\Lambda(\phi) = \Lambda(\phi + 2k\pi), \quad k \in \mathbb{Z}.
\]
In this case, straightforward calculations give
\[
E \Delta \tilde{\psi}_t \Delta \tilde{\psi}_t^T = \left( \frac{1}{\frac{X^o}{\omega}} \left[ \frac{X^o}{\omega^2} \right]^2 \left( \frac{1}{\frac{X^o}{\omega}} + \frac{1}{4} \right) \right), \tag{3.45}
\]
which is singular. Thus (3.40) is not valid in this case.

Example 3.1 and Example 3.2 give an indication about the need for further conditions to guarantee the inequality (3.40).
Assumptions 3.7-3.9 give sufficient requirements on the driving signal to guarantee (3.40). Assumption 3.7 is needed to simplify the analysis of Appendix 3.A. It simply means removing the bias from the periodic signal before applying the RPEM algorithm. Assumption 3.8 means that the periodic signal must be continuous and have a smooth curvature at the minimum or the maximum point of the signal. This assumption is needed to be able to use a Taylor series expansion around this minimum or maximum point. It can be noticed that Assumption 3.8 is not fulfilled for the periodic signal of Example 3.2. Assumption 3.9 is needed to evaluate the expectation in the inequality (3.40).

Remark 3.2. Note that in the rest of the analysis in this chapter, all the quantities given are evaluated at the true parameter vector \( \theta^o \) even when this is not stated explicitly. Hence, for notational convenience \( \theta \) is used instead of \( \theta^o \).

3.4 The Cramér-Rao Bound

In this section the CRB of the proposed parameterization is calculated. Introduce the following conditions:

Assumption 3.11. \( E [w(t)w(s)] = \sigma^2 \delta_{ts} \) and \( w(t) \) is Gaussian.

Assumption 3.12. \( N > N_o \) such that there exist a time instant \( t < N_o \) where \( \hat{u} \in I_j \) and \( \hat{w} \in [u_i^j, u_{i+1}^j] \) \( \forall i, j \in \{i = -k^j, \ldots, k^j - 1, j = 1, \ldots, L\} \).

Remark 3.3. Assumption 3.12 means that there is signal energy in each subinterval of the model, cf. [121] and Assumption 3.10.

Then the following theorem holds for the signal model of Assumption 3.1:

Theorem 3.2. Under Assumptions 2.1-2.2 and Assumptions 3.1-3.12, the CRB for \( (\theta_1^o \theta_\nu^o)^T \) is given by

\[
\text{CRB}(\theta) = \sigma^2 \left( \sum_{t=1}^{N} I(t) \right)^{-1},
\]

where

\[
I(t) = \begin{pmatrix}
I_{X,X} & I_{X,\omega} & I_{X,f_0} & 0 & \cdots & 0 & I_{X,t_1^j} & I_{X,t_1^j+1} & 0 & \cdots & 0 \\
I_{\omega,X} & I_{\omega,\omega} & I_{\omega,f_0} & 0 & \cdots & 0 & I_{\omega,t_1^j} & I_{\omega,t_1^j+1} & 0 & \cdots & 0 \\
I_{f_0,X} & I_{f_0,\omega} & I_{f_0,f_0} & 0 & \cdots & 0 & I_{f_0,t_1^j} & I_{f_0,t_1^j+1} & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & \cdots \\
\vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots & \vdots & \cdots & \cdots \\
I_{i_1^j,X} & I_{i_1^j,\omega} & I_{i_1^j,f_0} & 0 & \cdots & 0 & I_{i_1^j,t_1^j} & I_{i_1^j,t_1^j+1} & 0 & \cdots & 0 \\
i_{i_1^j+1,X} & I_{i_1^j+1,\omega} & I_{i_1^j+1,f_0} & 0 & \cdots & 0 & I_{i_1^j+1,t_1^j} & I_{i_1^j+1,t_1^j+1} & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & \cdots \\
\vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots & \vdots & \cdots & \cdots \\
0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & \cdots 
\end{pmatrix}.
\]
\( \hat{u}(t, \theta_i) \in [u_i^j, u_{i+1}^j] \in I_i, \ i = -k_j, \ldots, k_j - 1, \ j = 1, \ldots, L; \)

\[
\begin{align*}
I_{X,X} &= \left[ \frac{\partial f_j(\cdot)}{\partial u} \right]^2 \Lambda^2(\phi) \bigg|_{\phi = \omega t}, \\
I_{\omega,\omega} &= \left[ \frac{\partial f_j(\cdot)}{\partial u} \right]^2 X^2 \left[ \frac{d\Lambda(\phi)}{d\phi} \right]^2 \bigg|_{\phi = \omega t}, \\
I_{f_o,f_o} &= \left[ \frac{\partial f_j(\cdot)}{\partial u} \right]^2, \\
I_{X,\omega} &= \left[ \frac{\partial f_j(\cdot)}{\partial u} \right]^2 Xt \Lambda(\phi) \left[ \frac{d\Lambda(\phi)}{d\phi} \right] \bigg|_{\phi = \omega t}, \\
I_{X,f_i} &= \left[ \frac{\partial f_j(\cdot)}{\partial u} \right]^2 \Lambda(\phi) \bigg|_{\phi = \omega t}, \\
I_{X,f_{i+1}} &= \left[ \frac{\partial f_j(\cdot)}{\partial u} \right]^2 \Lambda(\phi) \bigg|_{\phi = \omega t}, \\
I_{\omega,f_i} &= \left[ \frac{\partial f_j(\cdot)}{\partial u} \right]^2 \Lambda(\phi) \bigg|_{\phi = \omega t}, \\
I_{\omega,f_{i+1}} &= \left[ \frac{\partial f_j(\cdot)}{\partial u} \right]^2 \Lambda(\phi) \bigg|_{\phi = \omega t}, \\
I_{f_o,f_i} &= \left[ \frac{\partial f_j(\cdot)}{\partial u} \right]^2 \left[ \frac{\partial f_j(\cdot)}{\partial f_{i+1}} \right]^2, \\
I_{f_o,f_{i+1}} &= \left[ \frac{\partial f_j(\cdot)}{\partial u} \right]^2 \left[ \frac{\partial f_j(\cdot)}{\partial f_{i+1}} \right]^2, \\
I_{f_i,f_{i+1}} &= \left[ \frac{\partial f_j(\cdot)}{\partial u} \right]^2 \left[ \frac{\partial f_j(\cdot)}{\partial f_{i+1}} \right]^2.
\end{align*}
\]

**Proof.** See Appendix 3.C.

**Remark 3.4.** The details of the proof follow [121] closely.
3. A Modified Nonlinear Approach to Periodic Signal Modeling

3.5 Numerical Examples

In order to study the performance of the modified RPEM algorithm suggested for joint estimation of the driving frequency and the parameters of the nonlinear output function, the following simulation examples were performed.

Example 3.3. Convergence to the true parameter vector.

The data were generated according to the following description: the driving wave was given by 

\[ u(t, \theta_o) = X_o \sin \omega_o t \]

where \( \omega_o = 2\pi \times 0.05 \) and \( X_o = 1 \). Two static nonlinearities \( (L = 2) \) were used as shown in Fig. 3.1;

\[ g_1 = \begin{pmatrix} -1 & -0.3 & -0.15 & 0.15 & 0.3 & 1 \end{pmatrix}^T, \]
\[ g_2 = \begin{pmatrix} -1 & -0.3 & 0.3 & 1 \end{pmatrix}^T, \]
\[ \theta_1 = \begin{pmatrix} -0.8 & -0.3 & 0.3 & 0.8 \end{pmatrix}^T, \quad u(t, \theta_1^o) \in I_1, \]
\[ \theta_2 = \begin{pmatrix} -0.8 & -0.5 & 0.5 & 0.8 \end{pmatrix}^T, \quad u(t, \theta_2^o) \in I_2. \]

(3.49)

where \( u(t, \theta_1^o) \in I_1 \) for positive slopes and \( u(t, \theta_2^o) \in I_2 \) for negative slopes, respectively. The additive noise was white zero-mean Gaussian with variance \( \sigma^2 = 0.01 \). Algorithm (3.9) was initialized with \( \lambda(0) = 0.95, \lambda_o = 0.99, P(0) = 0.01I, X = 1, K_0 = 1, f_0 = 0 \) and \( \omega(0) = 2\pi \times 0.02 \). Furthermore, the grid points in (3.49) were used, and the initial values for the nonlinearities were given by straight lines with unity slope.

The signal and the estimated signal model are given in Fig. 3.2(a). Also, the estimate of the driving frequency, the parameter estimates, and the prediction error are given in Figures 3.2(b)-3.2(e).
3.5. Numerical Examples

Figure 3.2: Convergence to the true parameter vector using Algorithm (3.9).
After 1000 samples the following estimates were obtained:

\[ \hat{X} = 0.95, \quad \hat{\omega} = 0.3141, \quad \hat{f}_o = -0.0094, \]
\[ \hat{\theta}_1 = \begin{pmatrix} -0.8705 & -0.2869 & 0.3024 & 0.8625 \end{pmatrix}^T, \]
\[ \hat{\theta}_2 = \begin{pmatrix} -0.8214 & -0.4910 & 0.5087 & 0.8280 \end{pmatrix}^T. \]

It can be concluded that the convergence to the true parameter vector is taking place.

**Example 3.4. Tracking fundamental frequency variations.**

As in Chapter 2, to improve the ability of the modified algorithm to track fundamental frequency variations, Algorithm (3.9) is modified to

\[ \varepsilon(t) = y(t) - \hat{y}(t) \]
\[ S(t) = \psi^T(t)P(t-1)\psi(t) + r_2(t) \]
\[ P(t) = P(t-1) - P(t-1)\psi(t)S^{-1}(t)\psi^T(t)P(t-1) + R_1(t) \]
\[ \begin{pmatrix} \hat{\theta}_1(t) \\ \hat{\theta}_n(t) \end{pmatrix} = \begin{pmatrix} \hat{\theta}_1(t-1) \\ \hat{\theta}_n(t-1) \end{pmatrix} + P(t)\psi(t)\varepsilon(t) \]

where \( R_1(t) \) and \( r_2(t) \) are the gain design variables.

The data were generated as in Example 3.3 and Algorithm (3.50) was initialized with \( P(0) = 10^{-3}I, \ X = 1, \ K_0 = 1, \ f_0 = 0 \) and \( \omega(0) = 2\pi \times 0.02 \). The design variables were \( R_1 = 10^{-4}I \) and \( r_2 = 0.25 \). Also, the grid points in (3.49) were used, and the initial values for the nonlinearities were given by straight lines with unity slope.

The signal and the estimated signal model are given in Fig. 3.3(a). Also, the true and estimated fundamental frequency are shown in Fig. 3.3(b). The parameter estimates and the prediction error are given in Figures 3.3(c)-3.3(e). After 2400 samples the following estimates were obtained:

\[ \hat{X} = 0.8432, \quad \hat{\omega} = 0.3474, \quad \hat{f}_o = -0.0499, \]
\[ \hat{\theta}_1 = \begin{pmatrix} -0.9169 & -0.2709 & 0.3477 & 0.9313 \end{pmatrix}^T, \]
\[ \hat{\theta}_2 = \begin{pmatrix} -0.8897 & -0.5422 & 0.4102 & 0.9036 \end{pmatrix}^T, \]

and it can be concluded that Algorithm (3.50) has the ability to track the fundamental frequency variations.

**Example 3.5. Modeling and tracking of flute music.**

In this example, a piece of flute music is modeled using Algorithm (3.50). The signal was extracted from a CD in .wav format with a sampling frequency of 22.05 kHz. The algorithm was initialized with \( P(0) = 500I, \ X = 1, \ K_0 = 1, \ f_0 = 0 \) and \( \omega(0) = 2\pi \times 348 \text{ rad/sec} \). The design variables were \( R_1 = 0.01I \) except that \( R_1(2, 2) = 0.05 \) to speed up frequency convergence. \( r_2 = 1 \) was used.
3.5. Numerical Examples

(a) The signal (dashed) and the estimated signal model (solid)

(b) Tracking the fundamental frequency

(c) Parameter estimates $\hat{\theta}_1$ and $\hat{f}_o$

(d) Parameter estimates $\hat{\theta}_2$

(e) Prediction error $\varepsilon$

Figure 3.3: Tracking the fundamental frequency of a sinusoid using Algorithm (3.50).
The grid points used for this example were
\[ g_1 = ( -0.12 - 0.06 - 0.03 - 0.015 - 0.0075 0.0075 0.015 0.03 0.06 0.12 )^T, \]
\[ g_2 = ( -0.12 - 0.06 - 0.03 - 0.015 0.015 0.03 0.06 0.12 )^T. \]
The results are given in Fig. 3.4. After 1000 samples the following estimates were obtained;
\[ \hat{X} = 0.1097, \quad \hat{\omega} = 2\pi(349.4114), \quad \hat{f}_0 = 0.0096, \]
\[ \hat{\theta}_1 = ( -0.1509 0.0144 0.0255 0.0155 0.0008 0.0057 0.0349 0.1156 )^T, \]
\[ \hat{\theta}_2 = ( -0.147 - 0.0266 - 0.028 - 0.0443 - 0.0368 - 0.0174 0.05 0.0964 )^T. \]
The plot of the estimated static nonlinearity is given in Fig. 3.5. Also, the comparison between the Hamming windowed periodogram with window width 400 samples for the data and the model for different number of the true parameters is given in Fig. 3.6. The results show that the method gives good results for different parameter vector sizes.

**Example 3.6. Performance of the RPE algorithm as compared to the CRB.**

In order to allow for comparison, the setup of this example closely resembles the numerical example of [121] and Example 2.4. In this example, 100 Monte-Carlo simulations were performed with different noise realizations to compare the performance of the modified algorithm (3.9) with the derived CRB for the fundamental frequency estimate. The data were generated and the algorithm was initialized as in Example 3.3. The statistics is based on excluding simulations that did not satisfy a margin of 5 standard deviations from the true fundamental frequency. Both the CRB for the fundamental frequency estimate and the MSE value were evaluated for data length of 2000 samples and for different SNR. The number of excluded simulations did not exceed 15 experiments. The statistical results are plotted in Fig. 3.7 which shows that the algorithm gives good statistical results.

### 3.6 Conclusions

The recursive periodic signal estimation algorithm of Chapter 2 has been modified by introducing an interval in the nonlinear block with fixed static gain. This modification allows a treatment of local convergence properties of the modified algorithm. It was proven that the algorithm is locally convergent to the true parameter vector by using the associated ODE approach. Then the modified algorithm was studied by numerical examples and it was shown that it can be easily modified to track fundamental frequency variations. Also, the CRB was calculated for the modified algorithm. Monte-Carlo experiments show that the modified algorithm gives good statistical results.
(a) The flute music (dashed) and the model output (solid)

(b) Tracking the fundamental frequency of the flute music

(c) Parameter estimates $\hat{\theta}_1$ and $\hat{f}_o$

(d) Parameter estimates $\hat{\theta}_2$

(e) Gain adaptation

(f) Prediction error $\varepsilon$

**Figure 3.4:** Modeling and tracking of the flute music using Algorithm (3.50).
Figure 3.5: Estimated static nonlinearity of the flute music. $f_1(u)$ (solid) and $f_2(u)$ (dashed).

Figure 3.6: The windowed periodogram for the flute music (solid) and the model (dashed).
3.A Proof of Lemma 3.2

Lemma 3.2 is proved here by following the analysis of [117, 118] closely. For notational convenience the dependence of the time $t$ is omitted in the derivation below.

Assuming that Assumption 3.4 is satisfied, $\psi_l$ can be written as

$$\psi_l = \frac{1}{\alpha_o} \psi^o_l + \Delta \psi_l,$$

(3.52)

where

$$\psi^o_l = \lim_{t \to \infty} E \, \text{gate}(\hat{u}) \, \psi_l,$$

(3.53)

and $\Delta \psi_l$ is the variation of $\psi_l$ around $\frac{1}{\alpha_o} \psi^o_l$. This implies

$$\lim_{t \to \infty} E \, \text{gate}(\hat{u}) \Delta \psi_l = 0.$$  

(3.54)

Now, the (1,1)-block of $\tilde{A}$ can be calculated, using Assumption 3.4, (3.52) and (3.54), as follows

$$\lim_{t \to \infty} E \, \text{gate}(\hat{u}) K_o^2 \psi_l \psi^T_l = \lim_{t \to \infty} E \, \text{gate}(\hat{u}) K_o^2 \left( \frac{1}{\alpha_o} \psi^o_l \psi^o_T + \frac{1}{\alpha_o} \psi^o_l \Delta \psi^T_l \
+ \frac{1}{\alpha_o} \Delta \psi^o_l \psi^o_T + \Delta \psi_l \Delta \psi^T_l \right) 
= K_o^2 \frac{1}{\alpha_o} \psi^o_l \psi^o_T + K_o^2 \lim_{t \to \infty} E \, \text{gate}(\hat{u}) \Delta \psi_l \Delta \psi^T_l.$$  

(3.55)

Thus $\tilde{A}$ can be written as

$$\tilde{A} = \alpha_o \left( \frac{K_o}{\alpha_o} \psi^o_l \right) \left( \frac{K_o}{\alpha_o} \psi^o_T \alpha_o \right) + \left( \lim_{t \to \infty} E \, \text{gate}(\hat{u}) K_o^2 \Delta \psi_l \Delta \psi^T_l \right) 0 
= \begin{pmatrix} F & M \\ M^T & H \end{pmatrix} + \begin{pmatrix} \tilde{F} & 0 \\ 0 & 0 \end{pmatrix}. $$

(3.56)
Applying Lemma 3.1 again and taking into account that $H = \alpha_o > 0$ by Assumption 3.4, positive definiteness of $\tilde{A}$ follows, if

$$\tilde{F} = \lim_{t \to \infty} E \text{gate}(\tilde{u}) K_o^2 \Delta \psi_l \Delta \psi_l^T > 0.$$  

In order to proceed, it is necessary to consider Assumption 3.5. This condition means that the contribution to the expectation from $I_o$, should not be negligible as compared to the whole contribution to the expectation. It is a condition on the amplitude distribution of the driving signal, i.e. the driving signal should be such that a sufficient amount of signal energy is located in the mid-subinterval $I_o$. Taking into account that $|K_o| > 0$ by Assumption 3.6, it remains to study when

$$\lim_{t \to \infty} E \Delta \psi_l \Delta \psi_l^T > 0$$

is satisfied. Since Eq. (3.54) does not imply that $\Delta \psi_l$ has zero mean, $\Delta \psi_l$ is written as

$$\Delta \psi_l = \Delta \psi_l^o + \Delta \tilde{\psi}_l,$$

where

$$\Delta \psi_l^o = \lim_{t \to \infty} E \Delta \psi_l.$$  

Then $\Delta \tilde{\psi}_l$ has zero mean, and

$$\lim_{t \to \infty} E \Delta \psi_l \Delta \psi_l^T = \Delta \psi_l^o \Delta \psi_l^o^T + \lim_{t \to \infty} E \Delta \tilde{\psi}_l \Delta \tilde{\psi}_l^T + \lim_{t \to \infty} E \Delta \psi_l \Delta \tilde{\psi}_l^T.$$  

Therefore, (3.40) is introduced. Since $\Delta \tilde{\psi}_l$ has zero mean, and

$$\psi_l = \frac{1}{\alpha_o} \psi_l^o + \Delta \psi_l^o + \Delta \tilde{\psi}_l,$$

by introducing Assumption 3.7, it follows that

$$\psi_l = \Delta \tilde{\psi}_l.$$  

Hence it is possible to analyze $\psi_l$ directly since for $\delta > 0$ it follows that

$$\tilde{F} = K_o^2 \lim_{t \to \infty} E \text{gate}(\tilde{u}) \Delta \psi_l \Delta \psi_l^T$$

$$\geq \delta K_o^2 \lim_{t \to \infty} E \Delta \psi_l \Delta \psi_l^T$$

$$\geq \delta K_o^2 \lim_{t \to \infty} E \Delta \tilde{\psi}_l \Delta \tilde{\psi}_l^T$$

$$= \delta K_o^2 \lim_{t \to \infty} E \psi_l \psi_l^T.$$  

(3.57)

So far, the treatment parallels that of [118]. However, from now on the difference of the linear block needs to be exploited, and the analysis becomes different from [118]. To find the conditions that guarantee (3.40), $\tilde{F}$ must be proved to be positive definite. It is suitable to expand $\Lambda(\phi)$ and $\frac{d\Lambda(\phi)}{d\phi}$ around $\phi_o$, where $\phi_o$ is a maximum or a minimum point for $\Lambda(\phi)$. This requires
the introduction of Assumption 3.8. The expansion, using properties of the maximum or minimum point, is

\[
\Lambda(\phi) = \Lambda(\phi_o) + \frac{1}{2} \frac{d^2 \Lambda(\phi)}{d\phi^2} \bigg|_{\phi=\phi_o} (\phi - \phi_o)^2 + O((\phi - \phi_o)^3),
\]

\[
\frac{d\Lambda(\phi)}{d\phi} = \frac{d^2 \Lambda(\phi)}{d\phi^2} \bigg|_{\phi=\phi_o} (\phi - \phi_o) + O((\phi - \phi_o)^2),
\]

where \( O \) denotes the ordo concept. It follows that, around this point

\[
\Lambda^2(\phi) = \left[ \Lambda(\phi_o) \right]^2 + \Lambda(\phi_o) \frac{d^2 \Lambda(\phi)}{d\phi^2} \bigg|_{\phi=\phi_o} (\phi - \phi_o)^2 + O((\phi - \phi_o)^3),
\]

\[
\left[ \frac{d\Lambda(\phi)}{d\phi} \right]^2 = \left[ \frac{d^2 \Lambda(\phi)}{d\phi^2} \bigg|_{\phi=\phi_o} \right]^2 (\phi - \phi_o)^2 + O((\phi - \phi_o)^3),
\]

\[
\Lambda(\phi) \frac{d\Lambda(\phi)}{d\phi} = \Lambda(\phi_o) \frac{d^2 \Lambda(\phi)}{d\phi^2} \bigg|_{\phi=\phi_o} (\phi - \phi_o) + O((\phi - \phi_o)^2).
\]

To guarantee that \( \tilde{F} > 0 \), it is required that (cf. Eq. (3.57))

\[
\tilde{F} \triangleq \lim_{t \to \infty} E \psi_t \psi_t^T = \lim_{t \to \infty} E \left( \sum_{\omega} \phi \Lambda(\phi_o) \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi=\phi_o} \frac{X^2}{\omega^2} \phi^2 \right) > 0.
\]

Now

\[
\lim_{t \to \infty} E(\cdot) = \frac{1}{2\pi} \int_0^{2\pi} (\cdot) h_\phi \, d\phi \geq \frac{1}{2\pi} \int_{\phi_o - \alpha}^{\phi_o + \alpha} (\cdot) h_\phi \, d\phi \triangleq \lim_{t \to \infty} E_\alpha(\cdot)
\]

for all positive semidefinite integrands. Restricting the consideration to the small interval \([\phi_o - \alpha, \phi_o + \alpha]\), \( \alpha \) arbitrary small, it follows that

\[
\tilde{F} \geq \lim_{t \to \infty} E_\alpha \left( \sum_{\omega} \phi \Lambda(\phi_o) \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi=\phi_o} \frac{X^2}{\omega^2} \phi^2 \right) > 0.
\]

Hence, it turns out that \( \tilde{F} > 0 \) if the right hand matrix of (3.63) is positive definite. Substituting by Eqs. (3.59)-(3.61) in Eq. (3.63), the minors of this matrix must satisfy the following inequalities

\[
\left[ \Lambda(\phi_o) \right]^2 + \Lambda(\phi_o) \frac{d^2 \Lambda(\phi)}{d\phi^2} \bigg|_{\phi=\phi_o} E_\alpha [(\phi - \phi_o)^2] > 0,
\]

\[
\left[ \frac{X^2}{\omega^2} \right]^2 \left[ \frac{d^2 \Lambda(\phi)}{d\phi^2} \bigg|_{\phi=\phi_o} \right]^2 E_\alpha [\phi^2 (\phi - \phi_o)^2] > 0,
\]

\[
\left[ \Lambda(\phi_o) \right]^2 + \Lambda(\phi_o) \frac{d^2 \Lambda(\phi)}{d\phi^2} \bigg|_{\phi=\phi_o} E_\alpha [(\phi - \phi_o)^2] E_\alpha [\phi^2 (\phi - \phi_o)^2] > 0
\]

\[
\left[ \Lambda(\phi_o) \right]^2 E_\alpha [\phi (\phi - \phi_o)]^2.
\]
Straightforward calculations for \( E_\alpha[(\phi - \phi_o)^2] \), \( E_\alpha[\phi^2(\phi - \phi_o)] \) and \( E_\alpha[\phi(\phi - \phi_o)] \) through the interval \([\phi_o - \alpha, \phi_o + \alpha]\) (where \( \alpha \) is a very small positive number) using Assumption 3.9 (which guarantees that the pdf of \( \phi \) is “almost” constant in the interval under consideration) give

\[
E_\alpha[(\phi - \phi_o)^2] = \frac{1}{\pi} \frac{\alpha^3}{3} + \mathcal{O}(\alpha^4),
\]

(3.67)

\[
E_\alpha[\phi^2(\phi - \phi_o)^2] = \frac{1}{\pi} \left( \frac{\alpha^5}{5} + \frac{\alpha^3}{3} \phi_o^2 \right) + \mathcal{O}(\alpha^6),
\]

(3.68)

\[
E_\alpha[\phi(\phi - \phi_o)] = \frac{1}{\pi} \frac{\alpha^3}{3} + \mathcal{O}(\alpha^4),
\]

(3.69)

Substituting by (3.67)-(3.68) in (3.64)-(3.66), respectively, and selecting \( \alpha \) arbitrarily small (neglecting \( \mathcal{O}(\alpha^4) \) etc.), it follows that

\[
\left[ \Lambda(\phi_o) \right]^2 + \Lambda(\phi_o) \frac{d^2 \Lambda(\phi)}{d\phi^2} \bigg|_{\phi = \phi_o} \frac{1}{\pi} \frac{\alpha^3}{3} > 0,
\]

(3.70)

\[
\left[ \frac{X_o^2}{\omega^2} \left[ \frac{d^2 \Lambda(\phi)}{d\phi^2} \bigg|_{\phi = \phi_o} \right]^2 \frac{1}{\pi} \frac{\alpha^3}{3} \phi_o^2 \right] > 0,
\]

(3.71)

\[
\left[ \Lambda(\phi_o) \right]^2 + \Lambda(\phi_o) \frac{d^2 \Lambda(\phi)}{d\phi^2} \bigg|_{\phi = \phi_o} \frac{1}{\pi} \frac{\alpha^3}{3} \left[ \frac{1}{\pi} \frac{\alpha^3}{3} \phi_o^2 \right] > 0,
\]

(3.72)

provided that

\[
|\Lambda(\phi_o)| \geq \epsilon_1 > 0,
\]

\[
\left| \frac{d^2 \Lambda(\phi)}{d\phi^2} \right|_{\phi = \phi_o} \geq \epsilon_2 > 0,
\]

(3.73)

\[
\phi_o \neq 0
\]

Note that \( \phi_o = \omega_o t \) cannot be equal to zero since \( \omega_o \neq 0 \) and time \( t \) is arbitrary chosen. Thus (3.62) is guaranteed for any continuous periodic signal with a minimum or a maximum point. The result follows.

### 3.8 Proof of Lemma 3.3

Following [117, 118], Lemma 3.2 can be proved as follows.

Recalling that the matrix \( C \) is defined as

\[
C = \lim_{t \to \infty} \mathbb{E} \left( (1 - \text{gate}(\tilde{u})) \begin{pmatrix} \frac{\partial f_1(\cdot)}{\partial \theta_m}^T & \frac{\partial f_1(\cdot)}{\partial \theta_1} & \cdots & \frac{\partial f_L(\cdot)}{\partial \theta_m}^T & \frac{\partial f_L(\cdot)}{\partial \theta_1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial f_1(\cdot)}{\partial \theta_m}^T & \frac{\partial f_1(\cdot)}{\partial \theta_L} & \cdots & \frac{\partial f_L(\cdot)}{\partial \theta_m}^T & \frac{\partial f_L(\cdot)}{\partial \theta_L} \end{pmatrix} \right),
\]

(3.74)

and noticing that

\[
\frac{\partial f_m(\cdot)}{\partial \theta_m}^T \frac{\partial f_n(\cdot)}{\partial \theta_n} = 0 \quad \text{if} \quad n \neq m
\]

(3.75)
3.B. Proof of Lemma 3.3

$C$ can be written as

\[
C = \lim_{t \to \infty} E \left( 1 - \text{gate}(\tilde{u}) \right) \begin{pmatrix}
\frac{\partial f_1(\cdot)}{\partial \theta_1} & \frac{\partial f_1(\cdot)}{\partial \theta_2} & \cdots & \frac{\partial f_1(\cdot)}{\partial \theta_r} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_L(\cdot)}{\partial \theta_1} & \frac{\partial f_L(\cdot)}{\partial \theta_2} & \cdots & \frac{\partial f_L(\cdot)}{\partial \theta_r}
\end{pmatrix}.
\] (3.76)

Since

\[
\frac{\partial f_j(\cdot)}{\partial \theta_j} = \mathbf{0}, \quad \tilde{u}(t, \theta^o) \in I_o
\] (3.77)

it follows that

\[
\lim_{t \to \infty} E \left( 1 - \text{gate}(\tilde{u}) \right) \frac{\partial f_j(\cdot)^T\partial f_j(\cdot)}{\partial \theta_j} = \lim_{t \to \infty} E \frac{\partial f_j(\cdot)^T\partial f_j(\cdot)}{\partial \theta_j}.
\] (3.78)

Hence

\[
C = \lim_{t \to \infty} E \begin{pmatrix}
\frac{\partial f_1(\cdot)}{\partial \theta_1} & \frac{\partial f_1(\cdot)}{\partial \theta_2} & \cdots & \frac{\partial f_1(\cdot)}{\partial \theta_r} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_L(\cdot)}{\partial \theta_1} & \frac{\partial f_L(\cdot)}{\partial \theta_2} & \cdots & \frac{\partial f_L(\cdot)}{\partial \theta_r}
\end{pmatrix}.
\] (3.79)

Also it can be noticed from the RPEM algorithm (3.9) that the $(j,j)$-block $(C_{jj})$ of the matrix $C$ can be written as

\[
C_{jj} = \begin{pmatrix}
C_{jj}^- & 0 \\
0 & C_{jj}^+
\end{pmatrix},
\] (3.80)

where $C_{jj}^-$ and $C_{jj}^+$ are band-matrices given by

\[
C_{jj}^- = \lim_{t \to \infty} E \begin{pmatrix}
\frac{\partial f_j}{\partial \theta_j} & \frac{\partial f_j}{\partial \theta_{j+1}} & \cdots & \frac{\partial f_j}{\partial \theta_{j+k_j-1}} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_j}{\partial \theta_{k_j-k_j-2}} & \frac{\partial f_j}{\partial \theta_{k_j-k_j-1}} & \cdots & \frac{\partial f_j}{\partial \theta_{k_j}}
\end{pmatrix},
\] (3.81)

\[
C_{jj}^+ = \lim_{t \to \infty} E \begin{pmatrix}
\frac{\partial f_j}{\partial \theta_j} & \frac{\partial f_j}{\partial \theta_{j+1}} & \cdots & \frac{\partial f_j}{\partial \theta_{j+k_j-1}} \\
0 & \vdots & \ddots & \vdots \\
0 & \frac{\partial f_j}{\partial \theta_{k_j-k_j-2}} & \frac{\partial f_j}{\partial \theta_{k_j-k_j-1}} & \cdots & \frac{\partial f_j}{\partial \theta_{k_j}}
\end{pmatrix},
\] (3.82)

If these two matrices are positive definite then also $C_{jj}$ is positive definite. Positive definiteness of $C_{jj}$ consequently proves positive definiteness of $C$. Since
\( C_{jj}^- \) and \( C_{jj}^+ \) have similar structure, it is sufficient to investigate one of them as done in [117, 118].

Investigate, for example, \( C_{jj}^+ \). Since \( C_{jj}^+ \) is positive semidefinite by construction, positive definiteness of \( C_{jj}^+ \) can be proved if

\[
 d^T C_{jj}^+ d = 0 \implies d = 0 \tag{3.83}
\]

where \( d \) is an arbitrary vector. Using Eq. (3.82) gives

\[
 d^T C_{jj}^+ d = \lim_{t \to \infty} E \left( \sum_{m=1}^{k_j^+} d_m \frac{\partial f_j}{\partial p_m} \right)^2, \tag{3.84}
\]

where \( d_m, m = 1, \ldots, k^+_j \) are the components of the vector \( d \). Let the function \( f_{C_{jj}^+}(d, \hat{u}) \) be given by

\[
 f_{C_{jj}^+}(d, \hat{u}) \triangleq \sum_{m=1}^{k^+_j} d_m \frac{\partial f_j}{\partial p_m} \neq 0, \quad \forall \hat{u} \in \left[ u^j_i, u^j_{i+1} \right]. \tag{3.85}
\]

This function represents a piecewise linear curve (see [118]) and its values in the grid points \( u^j_i, i = 1, \ldots, k^+_j \) are \( d_m, m = 1, \ldots, k^+_j \), i.e.

\[
 f_{C_{jj}^+}(d, u^j_i) = d_i, \quad i = 1, \ldots, k^+_j. \tag{3.86}
\]

This is because when \( \hat{u} \in \left[ u^j_i, u^j_{i+1} \right], i = 1, \ldots, k^+_j - 1 \), the function \( f_{C_{jj}^+}(d, \hat{u}) \) can be written as

\[
 f_{C_{jj}^+}(d, \hat{u}) = d_i \frac{u^j_{i+1} - \hat{u}}{u^j_{i+1} - u^j_i} + d_{i+1} \frac{\hat{u} - u^j_i}{u^j_{i+1} - u^j_i},
\]

which is a linear function of \( \hat{u} \) that satisfies

\[
 f_{C_{jj}^+}(d, u^j_i) = d_i,
\]

\[
 f_{C_{jj}^+}(d, u^j_{i+1}) = d_{i+1}.
\]

Using Assumption 3.10, Eq. (3.84) can be written as

\[
 d^T C_{jj}^+ d = \lim_{t \to \infty} E \left[ f_{C_{jj}^+}(d, \hat{u}) \right]^2
 = \int_{u^j_i}^{u^j_{i+1}} \left[ f_{C_{jj}^+}(d, \hat{u}) \right]^2 h_{\hat{u}}(\hat{u}) d\hat{u}. \tag{3.87}
\]

Now, conditions that imply (3.83) can be investigated. These conditions can be found by equating (3.87) to zero. Let \( I^j_i = (u^j_{i-1}, u^j_i), i = -k^-_j + 1, \ldots, k^+_j, \)

\[
 j = 1, \ldots, L \text{ denote the (open) subintervals of the piecewise linear model of the static nonlinearity including } I_0. \text{ Since by Assumption 3.10}
\]

\[
 h_{\hat{u}}(\hat{u}) \geq \delta_1 > 0 \tag{3.88}
\]
3.C. Proof of Theorem 3.2

in at least one nonzero interval \([a^j_i, b^j_i] \subset I^j_i\). Since \(\left[f_{C_{j,j^+}}(d, \tilde{u})\right]^2\) is nonnegative and continuous on \([u^j_i, u^j_{k^+}]\), it follows from equation (3.87) and Assumption 3.10 that

\[
f_{C_{j,j^+}}(d, \tilde{u}) = 0, \quad \tilde{u} \in [a_i, b_i] \subset I_{i}^j, \quad i = 1, \ldots, k^+ \quad j = 1, \ldots, L.
\]

Thus \(f_{C_{j,j^+}}(d, \tilde{u})\) is analytic in \(I_{i}^j\) and equals zero in nonzero intervals \([a_i, b_i] \subset I_{i}^j\). Hence, it follows from the properties of analytic functions (see e.g. [27]) that

\[
f_{C_{j,j^+}}(d, \tilde{u}) = 0, \quad \tilde{u} \in I_{i}^j, \quad i = 1, \ldots, k^+ \quad j = 1, \ldots, L.
\]

Since the function \(f_{C_{j,j^+}}(d, u_i^j)\) is continuous, Eq. (3.86) finally gives

\[
f_{C_{j,j^+}}(d, u_i^j) = d_i = 0, \quad i = 1, \ldots, k^+.
\]

This means that (3.83) is satisfied for all driving signals that fulfill Assumption 3.10.

The previous analysis proves positive definiteness of \(C_{j,j^+}\). Since \(C_{j,j^-}\) has similar structure as \(C_{j,j^+}\), \(C_{j,j}\) is also positive definite. This consequently leads to the positive definiteness of \(C\).

3.C Proof of Theorem 3.2

The proof of Theorem 3.2 follows the main lines of a similar result in [121]. The log-likelihood function \(l(\theta)\) of the RPEM algorithm (3.9) is given by

\[
l(\theta) = \text{constant} - \frac{1}{2\sigma^2} \sum_{t=1}^{N} (y(t) - \tilde{y}(t, \theta))^2,
\]  

(3.89)

Hence, the gradient of the log-likelihood function is defined as (cf. Eqs. (3.1) and (3.7))

\[
\frac{\partial l(\theta)}{\partial \theta} = \left( \frac{\partial l(\theta)}{\partial \theta_1}, \ldots, \frac{\partial l(\theta)}{\partial \theta_n} \right),
\]  

(3.90)

where

\[
\frac{\partial l(\theta)}{\partial \theta_1} = \left( \frac{\partial l(\theta)}{\partial X}, \frac{\partial l(\theta)}{\partial \omega} \right),
\]

\[
\frac{\partial l(\theta)}{\partial \theta_n} = \left( \frac{\partial l(\theta)}{\partial f_1}, \ldots, \frac{\partial l(\theta)}{\partial \theta_1}, \ldots, \frac{\partial l(\theta)}{\partial \theta_{L}} \right),
\]

(3.91)

\[
\frac{\partial l(\theta)}{\partial \theta_j} = \left( \frac{\partial l(\theta)}{\partial f_j^{1}}, \ldots, \frac{\partial l(\theta)}{\partial f_j^{1}}, \ldots, \frac{\partial l(\theta)}{\partial f_j^{k^+}} \right).
\]
In this case, the Fisher information matrix \([99]\) can be written as

\[
J = -E \frac{\partial l(\theta) T}{\partial \theta} \frac{\partial l(\theta)}{\partial \theta} = -E \left( \begin{array}{ccc}
\frac{\partial l(\theta) T}{\partial \theta} & \frac{\partial l(\theta)}{\partial \theta} & \frac{\partial l(\theta) T}{\partial \theta} \\
\frac{\partial l(\theta) T}{\partial \theta} & \frac{\partial l(\theta)}{\partial \theta} & \frac{\partial l(\theta) T}{\partial \theta} \\
\frac{\partial l(\theta) T}{\partial \theta} & \frac{\partial l(\theta)}{\partial \theta} & \frac{\partial l(\theta) T}{\partial \theta}
\end{array} \right).
\] (3.92)

In order to calculate the blocks of the matrix \(J\), the following relations are needed

\[
\begin{align*}
\frac{\partial l(\theta)}{\partial X} &= \frac{1}{\sigma^2} \sum_{t=1}^{N} (y(t) - \hat{y}(t, \theta)) \frac{\partial \hat{y}(t, \theta)}{\partial X}, \\
\frac{\partial l(\theta)}{\partial \omega} &= \frac{1}{\sigma^2} \sum_{t=1}^{N} (y(t) - \hat{y}(t, \theta)) \frac{\partial \hat{y}(t, \theta)}{\partial \omega}, \\
\frac{\partial l(\theta)}{\partial f_o} &= \frac{1}{\sigma^2} \sum_{t=1}^{N} (y(t) - \hat{y}(t, \theta)) \frac{\partial f_j(\cdot)}{\partial f_o}, \\
\frac{\partial l(\theta)}{\partial f_i} &= \frac{1}{\sigma^2} \sum_{t=1}^{N} (y(t) - \hat{y}(t, \theta)) \frac{\partial f_j(\cdot)}{\partial f_i}, \\
\frac{\partial l(\theta)}{\partial f_{i+1}} &= \frac{1}{\sigma^2} \sum_{t=1}^{N} (y(t) - \hat{y}(t, \theta)) \frac{\partial f_j(\cdot)}{\partial f_{i+1}},
\end{align*}
\] (3.93)

where

\[
\begin{align*}
\frac{\partial \hat{y}(t, \theta)}{\partial X} &= \frac{\partial f_j(\cdot)}{\partial u} \Lambda(\phi) |_{\phi=\omega t}, \\
\frac{\partial \hat{y}(t, \theta)}{\partial \omega} &= \frac{\partial f_j(\cdot)}{\partial u} X t \frac{d\Lambda(\phi)}{d\phi} |_{\phi=\omega t},
\end{align*}
\] (3.94)

for \(u(t_i) \in [u_i^j, u_{i+1}^j] \in I_j, \ i = -k_j^-, \ldots, k_j^+ - 1, \ j = 1, \ldots, L\). Thus using Assumption 3.1 and Assumption 3.11, it follows that

\[
\begin{align*}
E \left[ \frac{\partial^2 l(\theta)}{\partial X^2} \right] &= -\frac{1}{\sigma^2} \sum_{t=1}^{N} \left[ \frac{\partial f_j(\cdot)}{\partial u} \right]^2 \Lambda^2(\phi) |_{\phi=\omega t}, \\
E \left[ \frac{\partial^2 l(\theta)}{\partial \omega^2} \right] &= -\frac{1}{\sigma^2} \sum_{t=1}^{N} \left[ \frac{\partial f_j(\cdot)}{\partial u} \right]^2 X t \left \{ \frac{d\Lambda(\phi)}{d\phi} |_{\phi=\omega t} \right \}^2, \\
E \left[ \frac{\partial^2 l(\theta)}{\partial f_o^2} \right] &= \frac{1}{\sigma^2} \sum_{t=1}^{N} \left[ \frac{\partial f_j(\cdot)}{\partial f_o} \right]^2, \\
E \left[ \frac{\partial^2 l(\theta)}{\partial X \partial \omega} \right] &= \frac{1}{\sigma^2} \sum_{t=1}^{N} \left[ \frac{\partial f_j(\cdot)}{\partial u} \right]^2 X t \Lambda(\phi) \frac{d\Lambda(\phi)}{d\phi} |_{\phi=\omega t}, \\
E \left[ \frac{\partial^2 l(\theta)}{\partial \omega \partial f_o} \right] &= \frac{1}{\sigma^2} \sum_{t=1}^{N} \frac{\partial f_j(\cdot)}{\partial u} \frac{\partial f_j(\cdot)}{\partial f_o} \Lambda(\phi) |_{\phi=\omega t}, \\
E \left[ \frac{\partial^2 l(\theta)}{\partial \omega^2 \partial f_o} \right] &= \frac{1}{\sigma^2} \sum_{t=1}^{N} \frac{\partial f_j(\cdot)}{\partial u} X t \frac{d\Lambda(\phi)}{d\phi} |_{\phi=\omega t},
\end{align*}
\]
Taking into account that $\frac{\partial^2 l(\theta)}{\partial X \partial f_i}$, noticing that $J(3.48)$ and introducing the notation in (3.9). Thus the matrix $I(t)$, Eq. (3.46) directly follows from Assumption 3.12.

**Remark 3.5.** $\frac{\partial f_j(\cdot)}{\partial \theta}$ and $\frac{\partial f_j(\cdot)}{\partial f_l^i}$ can be calculated for different subintervals using (3.9). Thus the matrix $I(t)$ takes the following forms:

(i) When $\hat{u}(t, \theta_t) \in [u_i^j, u_{i+1}^j]$ \forall i = \{-k_1^-, \ldots, -2, 1, \ldots, k_1^+, \ldots, k_1^+, \ldots, \} \quad j = 1
-k_j^-, \ldots, k_j^+, \ldots, \} \quad j = 2, \ldots, L.$

$$I(t) = \begin{pmatrix}
I_{x,X} & I_{x,\omega} & 0 & \cdots & 0 & I_{x,\theta_{i+1}}^j & 0 & \cdots & 0
I_{\omega,X} & I_{\omega,\omega} & 0 & \cdots & 0 & I_{\omega,\theta_{i+1}}^j & 0 & \cdots & 0
0 & 0 & I_{\omega,\theta_{i+1}}^j & 0 & \cdots & 0 & \cdots & \cdots & \cdots
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \cdots & \cdots
0 & 0 & 0 & \cdots & I_{\omega,\omega} & 0 & \cdots & 0 & \cdots
0 & 0 & 0 & \cdots & 0 & I_{\omega,\theta_{i+1}}^j & 0 & \cdots & 0
0 & 0 & \cdots & \cdots & \cdots & \cdots & \ddots & \cdots & \cdots
0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & I_{x,\omega} & 0
0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 & I_{x,\theta_{i+1}}^j
0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 & 0
\end{pmatrix}, \quad (3.96)
where

\[
\begin{align*}
I_{X, X} &= \left( \frac{f_{i+1}^j - f_i^j}{u_{i+1}^j - u_i^j} \right)^2 \Lambda^2(\phi) \big|_{\phi = \omega t}, \\
I_{\omega, \omega} &= \left( \frac{f_{i+1}^j - f_i^j}{u_{i+1}^j - u_i^j} \right)^2 \frac{X^2 t^2}{d \Delta(\phi)} \bigg|_{\phi = \omega t}, \\
I_{X, \omega} &= \left( \frac{f_{i+1}^j - f_i^j}{u_{i+1}^j - u_i^j} \right)^2 X^t \Lambda(\phi) \frac{d \Delta(\phi)}{d \phi} \bigg|_{\phi = \omega t}, \\
I_{X, f_i^j} &= \frac{f_{i+1}^j - f_i^j}{u_{i+1}^j - u_i^j} \left( u_{i+1}^j - u_i^j - u(t, \theta_i) \right) \Lambda(\phi) \big|_{\phi = \omega t}, \\
I_{X, f_{i+1}^j} &= \frac{f_{i+1}^j - f_i^j}{u_{i+1}^j - u_i^j} \left( u_{i+1}^j - u_i^j - u(t, \theta_i) \right) \Lambda(\phi) \big|_{\phi = \omega t}, \\
I_{\omega, f_i^j} &= \frac{f_{i+1}^j - f_i^j}{u_{i+1}^j - u_i^j} \left( u_{i+1}^j - u_i^j - u(t, \theta_i) \right) X^t \frac{d \Delta(\phi)}{d \phi} \bigg|_{\phi = \omega t}, \\
I_{\omega, f_{i+1}^j} &= \frac{f_{i+1}^j - f_i^j}{u_{i+1}^j - u_i^j} \left( u_{i+1}^j - u_i^j - u(t, \theta_i) \right) X^t \frac{d \Delta(\phi)}{d \phi} \bigg|_{\phi = \omega t}, \\
I_{f_i^j, f_i^j} &= \left( \frac{u_{i+1}^j - u(t, \theta_i)}{u_{i+1}^j - u_i^j} \right)^2, \\
I_{f_{i+1}^j, f_{i+1}^j} &= \left( \frac{u(t, \theta_i) - u_{i+1}^j}{u_{i+1}^j - u_i^j} \right)^2, \\
I_{f_i^j, f_{i+1}^j} &= \frac{u_{i+1}^j - u(t, \theta_i)}{u_{i+1}^j - u_i^j} \left( u(t, \theta_i) - u_{i+1}^j \right).
\end{align*}
\]  

(ii) When \( u(t, \theta_i) \in [u_{i-1}^j, u_i^j] \subset I_i; 

\[
I(t) = \begin{pmatrix}
I_{X, X} & I_{X, \omega} & I_{X, f_i^j} & 0 & \cdots & 0 & I_{X, f_{i+1}^j} & 0 & \cdots & 0 \\
I_{\omega, X} & I_{\omega, \omega} & I_{\omega, f_i^j} & 0 & \cdots & 0 & I_{\omega, f_{i+1}^j} & 0 & \cdots & 0 \\
I_{f_i^j, X} & I_{f_i^j, \omega} & I_{f_i^j, f_i^j} & 0 & \cdots & 0 & I_{f_i^j, f_{i+1}^j} & 0 & \cdots & 0 \\
0 & 0 & 0 & \ddots & \cdots & \ddots & \ddots & \ddots & \cdots & \ddots \\
0 & 0 & 0 & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ddots & \ddots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ddots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix},
\]  

(3.98)
where

\[
\begin{align*}
I_{X,X} &= \left[ K_\omega u_\omega - f_o - f_1 \right]^2 - u_\omega - u_{-1}^1 \Lambda^2(\phi) |_{\phi=\omega t}, \\
I_{\omega,\omega} &= \left[ K_\omega u_\omega - f_o - f_1 \right]^2 - u_\omega - u_{-1}^1 \chi^2 \left[ \frac{d\Lambda(\phi)}{d\phi} \right] |_{\phi=\omega t}, \\
I_{f_o,f_o} &= \left[ u(t, \theta) - u_{-1}^1 \right]^2, \\
I_{X,\omega} &= \left[ K_\omega u_\omega - f_o - f_1 \right]^2 - u_\omega - u_{-1}^1 X_t \Lambda(\phi) \left| \frac{d\Lambda(\phi)}{d\phi} \right|_{\phi=\omega t}, \\
I_{X,f_o} &= \left[ K_\omega u_\omega - f_o - f_1 \right]^2 - u_\omega - u_{-1}^1 X_t \Lambda(\phi) \left| \frac{d\Lambda(\phi)}{d\phi} \right|_{\phi=\omega t}, \\
I_{\omega,f_o} &= \left[ K_\omega u_\omega - f_o - f_1 \right]^2 - u_\omega - u_{-1}^1 X_t \Lambda(\phi) \left| \frac{d\Lambda(\phi)}{d\phi} \right|_{\phi=\omega t}, \\
I_{f_o,f_o} &= \left[ u(t, \theta) - u_{-1}^1 \right]^2, \\
I_{f_1,f_1} &= \left[ u(t, \theta) - u_{-1}^1 \right]^2.
\end{align*}
\]

(iii) When \( \bar{u}(t, \theta) \in \[u_\omega^-, u_\omega^+\] \subset I_1:

\[
I(t) = \begin{pmatrix} I_{X,X} & I_{X,\omega} & I_{X,f_o} & 0 & \cdots & 0 \\
I_{\omega,X} & I_{\omega,\omega} & I_{\omega,f_o} & 0 & \cdots & 0 \\
I_{f_o,X} & I_{f_o,\omega} & I_{f_o,f_o} & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 0 \\
\end{pmatrix},
\]

where

\[
\begin{align*}
I_{X,X} &= K_\omega^2 \Lambda^2(\phi) |_{\phi=\omega t}, \\
I_{\omega,\omega} &= K_\omega^2 \chi^2 \left[ \frac{d\Lambda(\phi)}{d\phi} \right] |_{\phi=\omega t}, \\
I_{f_o,f_o} &= 1, \\
I_{X,\omega} &= K_\omega X_t \Lambda(\phi) \left| \frac{d\Lambda(\phi)}{d\phi} \right|_{\phi=\omega t}, \\
I_{X,f_o} &= K_\omega X_t \Lambda(\phi) \left| \frac{d\Lambda(\phi)}{d\phi} \right|_{\phi=\omega t}, \\
I_{\omega,f_o} &= K_\omega X_t \Lambda(\phi) \left| \frac{d\Lambda(\phi)}{d\phi} \right|_{\phi=\omega t}.
\end{align*}
\]
(iv) When \( \hat{u}(t, \theta_t) \in [u_{o+}, u_1] \subset I_1; \\
\begin{align*}
I(t) &=
\begin{pmatrix}
I_{X,X} & I_{X,\omega} & I_{X,f_0} & 0 & \cdots & 0 & I_{X,t_1} & 0 & \cdots & 0 \\
I_{\omega,X} & I_{\omega,\omega} & I_{\omega,f_0} & 0 & \cdots & 0 & I_{\omega,t_1} & 0 & \cdots & 0 \\
I_{f_0,X} & I_{f_0,\omega} & I_{f_0,f_0} & 0 & \cdots & 0 & I_{f_0,t_1} & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 0 & I_{f_0,t_1} & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & 0
\end{pmatrix},
\end{align*}
\tag{3.102}
\end{align*}

where
\begin{align*}
I_{X,X} &= \left[ \frac{f_1^1 - K_o u_{o+} - f_o}{u_1^1 - u_{o+}} \right]^2 \Lambda^2(\phi) \bigg|_{\phi=\omega t}, \\
I_{\omega,\omega} &= \left[ \frac{f_1^1 - K_o u_{o+} - f_o}{u_1^1 - u_{o+}} \right]^2 X^2 l^2 \left[ \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi=\omega t} \right]^2, \\
I_{f_0,f_0} &= \left[ \frac{u_1^1 - u(t, \theta_t)}{u_1^1 - u_{o+}} \right]^2, \\
I_{X,\omega} &= \left[ \frac{f_1^1 - K_o u_{o+} - f_o}{u_1^1 - u_{o+}} \right]^2 X t \Lambda(\phi) \left[ \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi=\omega t} \right], \\
I_{f_0,\omega} &= \left[ \frac{f_1^1 - K_o u_{o+} - f_o}{u_1^1 - u_{o+}} \right]^2 \Lambda(\phi) \bigg|_{\phi=\omega t}, \\
I_{X,f_0} &= \left[ \frac{f_1^1 - K_o u_{o+} - f_o}{u_1^1 - u_{o+}} \right]^2 \left[ \frac{u_1^1 - u(t, \theta_t)}{u_1^1 - u_{o+}} \right] \Lambda(\phi) \bigg|_{\phi=\omega t}, \\
I_{\omega,f_0} &= \left[ \frac{f_1^1 - K_o u_{o+} - f_o}{u_1^1 - u_{o+}} \right]^2 \left[ \frac{u_1^1 - u(t, \theta_t)}{u_1^1 - u_{o+}} \right] X t \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi=\omega t}, \\
I_{X,t_1} &= \left[ \frac{f_1^1 - K_o u_{o+} - f_o}{u_1^1 - u_{o+}} \right]^2 \left[ \frac{u_1^1 - u(t, \theta_t)}{u_1^1 - u_{o+}} \right] \Lambda(\phi) \bigg|_{\phi=\omega t}, \\
I_{\omega,t_1} &= \left[ \frac{f_1^1 - K_o u_{o+} - f_o}{u_1^1 - u_{o+}} \right]^2 \left[ \frac{u_1^1 - u(t, \theta_t)}{u_1^1 - u_{o+}} \right] X t \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi=\omega t}, \\
I_{f_0,t_1} &= \left[ \frac{u_1^1 - u(t, \theta_t)}{u_1^1 - u_{o+}} \right]^2, \\
I_{f_1^1, f_1^1} &= \left[ \frac{u(t, \theta_t) - u_{o+}}{u_1^1 - u_{o+}} \right]^2.
\end{align*}
Chapter 4

An Adaptive Grid Point Algorithm for Periodic Signal Modeling

4.1 Introduction

In the previous two chapters the nonlinearity was chosen to be piecewise linear with the estimated parameters being the function values in a fixed set of grid points, resulting in fixed grid point adaptation. In this chapter, the RPEM algorithm introduced in Chapter 3 is modified to enable the algorithm to estimate the grid points as well as the driving frequency and the parameters of the nonlinear output function, resulting in automatic grid point adaptation. This is expected to reduce the modeling errors since it gives the algorithm more freedom to choose the suitable grid points.

The contributions of this chapter can hence be summarized as follows. The algorithm (3.9) is modified to adaptively estimate the grid points. Also, the CRB is derived for the suggested method. Furthermore, the performance for joint estimation of the driving frequency, the parameters of the nonlinear output function as well as the grid points, is studied by numerical examples. The purpose is to investigate convergence to the true parameter vector, the ability to track both the fundamental frequency and the amplitude variations, and the performance as compared to the fixed grid point algorithm.

The chapter is organized as follows. In Section 4.2, the suggested algorithm is introduced. The derivation of the CRB for the suggested algorithm is presented in Section 4.3. Finally, Numerical examples and conclusions are given in Section 4.4 and Section 4.5, respectively.

4.2 The Suggested Algorithm

Similarly as done in Chapter 3 the driving input signal \( \hat{u}(t, \theta_l) \) is modeled as

\[
\hat{u}(t, \theta_l) = X \Lambda(\omega t), \\
\theta_l = \begin{bmatrix} X & \omega \end{bmatrix}^T,
\]

where \( t \) denotes discrete time, \( \omega \in [0, \pi] \) denotes the unknown normalized angular frequency: \( \omega = 2\pi f/f_s \) where \( f \) is the frequency, and \( f_s \) is the sampling frequency. \( X \) is a possibly time-varying parameter.
The piecewise linear model discussed in Chapter 3 is used also here for the parameterization of the nonlinearity. Choosing $I_o$ to be contained in the first interval $I_1$, the grid points are defined as

$$g_j^T = \begin{cases} (u_{-k_j^1}^1 \quad u_{-k_j^1+1}^1 \cdots u_{-k_j^1+I_o^1}^1 \quad u_{-k_j^1-1}^1 \quad u_{-k_j^1+I_o^1}^1), & j = 1 \\ (u_{-k_j^2}^2 \quad u_{-k_j^2+1}^2 \cdots u_{-k_j^2+I_o^2}^2 \quad u_{-k_j^2-1}^2 \quad u_{-k_j^2+I_o^2}^2), & j = 2, \cdots, L. \end{cases}$$

(4.2)

Then with $f_j(\theta_j, g_j, \hat{u}(t, \theta_i))$ denoting the nonlinearity to be used in $I_j$, the parameters $\theta_j$ are chosen as the values of $f_j(\theta_j, g_j, \hat{u}(t, \theta_i))$ in the grid points, i.e.

$$\theta_j = \left( f_{j-k_j^1}^1 \cdots f_{j-1}^1 \quad f_j^1 \cdots f_{k_j^1}^1 \right)^T, \quad j = 1, \cdots, L$$

$$f_j(\theta_j, g_j, u_j^1) = \begin{cases} K_o \hat{u}(t, \theta_i) + f_o, & \hat{u}(t, \theta_i) \in I_o, \quad j = 1 \\ f_i^j, & i = -k_j^1, \cdots, -1,1, \cdots, k_j^1, \quad j = 1, \cdots, L. \end{cases}$$

(4.3)

Also, here $K_o$ is a constant static gain chosen by the user and there are no parameters corresponding to $u_o^-$ and $u_o^+$ (cf. Eq. (3.6)). Therefore the model output becomes

$$\hat{y}(t, \theta) = f_j(\theta_j, g_j, \hat{u}(t, \theta_i)), \quad \hat{u}(t, \theta_i) \in I_j, \quad j = 1, \cdots, L$$

$$\theta = \left( \theta_n^T \quad \theta_n^T \quad \theta_n^T \right)^T,$$

$$\theta_i = \left( X \quad \omega \right)^T,$$

$$\theta_n = \left( f_o \quad \hat{\theta}_n^T \right)^T,$$

$$\hat{\theta}_n = \left( \theta_n^T \cdots \theta_n^T \right)^T,$$

$$\theta_g = \left( g_1^T \cdots g_L^T \right)^T.$$  

(4.4)

Hence, similarly as done in the previous two chapter, a recursive Gauss-Newton PEM follows by the minimization of the following cost function

$$V(\theta) = \lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} E[\varepsilon^2(t, \theta)].$$

(4.5)

The objective of this chapter is, as stated in the introduction, to estimate the grid points recursively in addition to the estimation of the fundamental frequency and the parameters of the nonlinear output function. This will give the algorithm more freedom to choose the grid points and achieve a better performance. In this case, the negative gradient of $\hat{y}(t, \theta)$ is given by

$$\psi(t, \theta) = \begin{pmatrix} \frac{\partial f_1(\cdot)}{\partial u} \psi_l(t) & \frac{\partial f_j(\cdot)}{\partial u} \cdots 0 \cdots 0 & \frac{\partial f_j(\cdot)}{\partial \theta} \cdots 0 \cdots 0 \frac{\partial f_j(\cdot)}{\partial \theta} \cdots 0 \cdots 0 \end{pmatrix}^T,$$

(4.6)

where

$$\psi_l(t) = \left( \Lambda(\phi) \left|_{\phi=\omega t} \right. \quad X t \frac{d \Lambda(\phi)}{d \phi} \right|_{\phi=\omega t} \right)^T.$$  

(4.7)
Thus the RPEM algorithm becomes

\[ \varepsilon(t) = y(t) - \hat{y}(t) \]
\[ \lambda(t) = \lambda_o \lambda(t - 1) + 1 - \lambda_o \]
\[ S(t) = \psi^T(t) P(t - 1) \psi(t) + \lambda(t) \]
\[ P(t) = (P(t - 1) - P(t - 1) \psi(t) S^{-1}(t) \psi^T(t) P(t - 1)) / \lambda(t) \]

\[
\begin{pmatrix}
\hat{\theta}_i(t) \\
\hat{\theta}_o(t) \\
\hat{\theta}_g(t)
\end{pmatrix}
= \left[ \begin{pmatrix}
\hat{\theta}_i(t - 1) \\
\hat{\theta}_o(t - 1) \\
\hat{\theta}_g(t - 1)
\end{pmatrix} + P(t) \psi(t) \varepsilon(t) \right]_{DM} (4.8)
\]

\[ \hat{u}(t + 1) = \hat{X}(t) \Lambda(\phi) \left. \right|_{\phi = \hat{u}(t) + 1} \]

\[ \psi_i(t + 1) = \left( \Lambda(\phi) \left. \right|_{\phi = \hat{u}(t) + 1} \hat{X}(t)(t + 1) \frac{d \Lambda(\phi)}{d \phi} \left. \right|_{\phi = \hat{u}(t) + 1} \right)^T \]

when \( \hat{u}(t + 1) \in I_1 \)

when \( \hat{u}(t + 1) \in [\hat{u}^-_1, u_o^-] \)

\[ \hat{y}(t + 1) = \hat{f}_1 + u_o^- - (K_o u_o^- + \hat{f}_o) \hat{u}^-_1 + \frac{K_o}{u_o^- - \hat{u}^-_1} \hat{u}(t + 1) \]
\[ \frac{\partial f_1}{\partial u} = \frac{\partial f_1}{\partial v} = \frac{K_o u_o^- + \hat{f}_o}{u_o^- - \hat{u}^-_1} \]
\[ \frac{\partial f_1}{\partial \hat{u}} = \frac{\partial f_1}{\partial \hat{v}} = \frac{\hat{u}(t + 1) - \hat{u}^-_1}{u_o^- - \hat{u}^-_1} \]
\[ \frac{\partial f_1}{\partial \hat{f}_o} = 0, \quad l \neq -1 \]
\[ \frac{\partial f_1}{\partial \hat{f}_l} = 0, \quad l \neq -1, 0 \]

end

when \( \hat{u}(t + 1) \in [u_o^-, u_o^+] \)

\[ \hat{y}(t + 1) = K_o \hat{u}(t + 1) + \hat{f}_o \]
\[ \frac{\partial f_1}{\partial u} = K_o \]
\[ \frac{\partial f_1}{\partial \hat{u}} = 1 \]
\[ \frac{\partial f_1}{\partial \hat{f}_o} = 0, \quad l \neq 0 \]
\[ \frac{\partial f_1}{\partial \hat{f}_l} = 0, \quad \forall l \]

end
when \( \tilde{u}(t + 1) \in [u_{o+1}, u_{o+2}] \)

\[
\tilde{y}(t + 1) = \left( K_o u_{o+1} + \tilde{f}_o \right) \tilde{u}_{i+1} \left( t \right) - \tilde{f}_1 u_{o+1} + \frac{\tilde{f}_1}{\tilde{u}_{i+1} - u_{o+1}} \tilde{y}(t + 1)
\]

\[
\frac{\partial f_1}{\partial \tilde{u}} = \frac{\tilde{f}_1}{\tilde{u}_{i+1} - u_{o+1}}
\]

\[
\frac{\partial f_1}{\partial \tilde{f}_1} = \frac{\tilde{f}_1}{\tilde{u}_{i+1} - u_{o+1}}
\]

\[
\frac{\partial f_1}{\partial \tilde{f}_1} = \tilde{u}_{i+1} - \tilde{u}(t + 1)
\]

\[
\frac{\partial f_1}{\partial \tilde{u}} = \tilde{u}_{i+1} - u_{o+1}
\]

\[
\frac{\partial f_1}{\partial \tilde{f}_1} = 0, \quad l \neq 1
\]

\[
\frac{\partial f_1}{\partial \tilde{f}_1} = 0, \quad l \neq 0, 1
\]

end

when \( \tilde{u}(t + 1) \in I_j, \quad j = 1, \ldots, L \)

when \( \tilde{u}(t + 1) \in [\tilde{u}_{i}, \tilde{u}_{i+1}] \quad \forall i = \left\{ -k_1, \ldots, -2, 1, \ldots, k_j - 1, \quad j = 1, \right. \left. -k_j, \ldots, k_j - 1, \quad j = 2, \ldots, L, \right. \)

\[
\tilde{y}(t + 1) = \left( \tilde{f}_1 \tilde{u}_{i+1} - \tilde{f}_i \tilde{u}_{i} \right) \frac{\tilde{u}_{i+1} - \tilde{u}_{i}}{\tilde{u}_{i+1} - \tilde{u}_{i}} + \frac{\tilde{f}_1 \tilde{u}_{i+1} - \tilde{f}_i \tilde{u}_{i}}{\tilde{u}_{i+1} - \tilde{u}_{i}} \tilde{y}(t + 1)
\]

\[
\frac{\partial f_1}{\partial \tilde{u}} = \frac{\tilde{f}_1 \tilde{u}_{i+1} - \tilde{f}_i \tilde{u}_{i}}{\tilde{u}_{i+1} - \tilde{u}_{i}}
\]

\[
\frac{\partial f_1}{\partial \tilde{f}_1} = \frac{\tilde{f}_1 \tilde{u}_{i+1} - \tilde{f}_i \tilde{u}_{i}}{\tilde{u}_{i+1} - \tilde{u}_{i}}
\]

\[
\frac{\partial f_1}{\partial \tilde{f}_1} = \tilde{u}_{i+1} - \tilde{u}(t + 1)
\]

\[
\frac{\partial f_1}{\partial \tilde{u}} = \tilde{u}_{i+1} - \tilde{u}_{i}
\]

\[
\frac{\partial f_1}{\partial \tilde{f}_1} = \tilde{u}_{i+1} - \tilde{u}_{i}
\]

\[
\frac{\partial f_1}{\partial \tilde{f}_1} = \tilde{u}_{i+1} - \tilde{u}_{i}
\]

\[
\frac{\partial f_1}{\partial \tilde{f}_1} = \tilde{u}_{i+1} - \tilde{u}_{i}
\]

\[
\frac{\partial f_1}{\partial \tilde{f}_1} = \tilde{u}_{i+1} - \tilde{u}_{i}
\]

\[
\frac{\partial f_1}{\partial \tilde{f}_1} = \tilde{u}_{i+1} - \tilde{u}_{i}
\]

\[
\frac{\partial f_1}{\partial \tilde{f}_1} = 0, \quad l \neq i, i + 1
\]

\[
\frac{\partial f_1}{\partial \tilde{f}_1} = 0
\]

\[
\frac{\partial f_1}{\partial \tilde{f}_1} = 0, \quad l \neq i, i + 1
\]

end
4.3. The Cramér-Rao Bound

\[ \frac{\partial f_i}{\partial \theta_j} = \left( \frac{\partial f_1}{\partial \theta_j} \ldots \frac{\partial f_k}{\partial \theta_j} \right), \quad \frac{\partial f_i}{\partial g_j} = \left( \frac{\partial f_1}{\partial g_j} \ldots \frac{\partial f_k}{\partial g_j} \right) \]

\[ \psi(t+1) = \left( \frac{\partial f_i}{\partial \theta_j} \psi_t(t+1) \frac{\partial f_{i+1}}{\partial \theta_j} \ldots \frac{\partial f_{i+k}}{\partial \theta_j} \frac{\partial f_{i+k+1}}{\partial \theta_j} \right)^T \]

Rem. 4.1. It is assumed here that some precautions are taken to prevent grid miss ordering during the estimation process. This is stated as

Assumption 4.1. Grid ordering is included in the definition of the model set.

4.3 The Cramér-Rao Bound

Theorem 4.1. Under Assumptions 2.1-2.2, Assumption 3.1, Assumptions 3.11-3.12 and Assumption 4.1, the CRB for \((\theta^*_1, \theta^*_2, \theta^*_3)^T\) is given by

\[ \text{CRB} (\theta) = \sigma^2 \left( \sum_{t=1}^{N} I(t) \right)^{-1} \]

where

\[ I(t) = \]

\[ \left( \begin{array}{cccccccc}
I_{X,1} & I_{X,2} & 0 & \ldots & 0 & I_{X,f_1} & 0 & \ldots & 0 \\
I_{X,1} & I_{X,2} & 0 & \ldots & 0 & I_{X,f_1} & 0 & \ldots & 0 \\
I_{X,1} & I_{X,2} & 0 & \ldots & 0 & I_{X,f_1} & 0 & \ldots & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\
\end{array} \right) \]

\[ \hat{u}(t, \theta_t) \in [u^i, u^i_{i+1}] \in I_j, i = -k_j^{-1}, \ldots, k_j^{-1} - 1, j = 1, \ldots, L. \]
\[
I_{X,X} = \left( \frac{\partial f_j(\cdot)}{\partial u} \right)^2 \Lambda^2(\phi) \big|_{\phi=\omega t},
\]
\[
I_{\omega,\omega} = \left( \frac{\partial f_j(\cdot)}{\partial u} \right)^2 X^2 t^2 \left[ \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi=\omega t} \right]^2,
\]
\[
I_{f_0,f_0} = \left( \frac{\partial f_j(\cdot)}{\partial f_0} \right)^2,
\]
\[
I_{X,\omega} = \left( \frac{\partial f_j(\cdot)}{\partial u} \right)^2 X t \Lambda(\phi) \frac{d\Lambda(\phi)}{d\phi} \big|_{\phi=\omega t},
\]
\[
I_{X,f_0} = \frac{\partial f_j(\cdot)}{\partial u} \frac{\partial f_j(\cdot)}{\partial f_0} \Lambda(\phi) \big|_{\phi=\omega t},
\]
\[
I_{X,f_i} = \frac{\partial f_j(\cdot)}{\partial u} \frac{\partial f_j(\cdot)}{\partial f_i} \Lambda(\phi) \big|_{\phi=\omega t},
\]
\[
I_{X,f_{i+1}} = \frac{\partial f_j(\cdot)}{\partial u} \frac{\partial f_j(\cdot)}{\partial f_{i+1}} \Lambda(\phi) \big|_{\phi=\omega t},
\]
\[
I_{X,u_i} = \frac{\partial f_j(\cdot)}{\partial u} \frac{\partial f_j(\cdot)}{\partial u_i} \Lambda(\phi) \big|_{\phi=\omega t},
\]
\[
I_{X,u_{i+1}} = \frac{\partial f_j(\cdot)}{\partial u} \frac{\partial f_j(\cdot)}{\partial u_{i+1}} \Lambda(\phi) \big|_{\phi=\omega t},
\]
\[
I_{\omega,f_i} = \frac{\partial f_j(\cdot)}{\partial u} \frac{\partial f_j(\cdot)}{\partial f_i} X t \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi=\omega t},
\]
\[
I_{\omega,f_{i+1}} = \frac{\partial f_j(\cdot)}{\partial u} \frac{\partial f_j(\cdot)}{\partial f_{i+1}} X t \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi=\omega t},
\]
\[
I_{\omega,u_i} = \frac{\partial f_j(\cdot)}{\partial u} \frac{\partial f_j(\cdot)}{\partial u_i} X t \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi=\omega t},
\]
\[
I_{\omega,u_{i+1}} = \frac{\partial f_j(\cdot)}{\partial u} \frac{\partial f_j(\cdot)}{\partial u_{i+1}} X t \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi=\omega t},
\]
\[
(4.11)
\]
4.4 Numerical Examples

In order to study the performance of the RPEM algorithm suggested for joint estimation of the driving frequency and the parameters of the nonlinear output function in a set of grid points estimated recursively, the following simulation examples were performed.


The data were generated according to the following description: the driving wave was given by \( u(t, \theta_0) = \sin \omega_0 t \) where \( \omega_0 = 2\pi \times 0.05 \). The static nonlinearity was chosen as

\[
 f(u) = \begin{cases} 
 (5/3)u^2 + 0.15, & u \geq 0.3 \\
 u, & -0.3 \leq u < 0.3 \\
 -(5/3)u^2 - 0.15, & u < -0.3 
\end{cases} 
\] (4.12)

Note that this means the system is not in the model set, i.e. undermodeling effects are at hand.

Proof. See Appendix 4.A. \( \square \)
Algorithm (4.8) was initialized with \( \lambda(0) = 0.95, \lambda_o = 0.99, P(0) = 0.01I, \) \( X = 1, K_o = 1, f_o = 0 \) and \( \omega(0) = 2\pi \times 0.02. \) The additive noise was white zero-mean Gaussian with variance \( \sigma^2 = 0.01. \) Further, two static nonlinearities \((L = 2)\) were used, where \( \hat{u}(t, \theta_l) \in I_1 \) for positive slopes and \( \hat{u}(t, \theta_l) \in I_2 \) for negative slopes, respectively. The nonlinearities were initialized as straight lines with unity slope in the following grid points:

\[
g_1(0) = \begin{bmatrix} -2 & -1 & -0.3 & -0.1 & 0.1 & 0.3 & 1 & 2 \end{bmatrix}^T, \quad (4.13)
g_2(0) = \begin{bmatrix} -2 & -1 & -0.3 & 0.3 & 1 & 2 \end{bmatrix}^T.
\]

The signal and the estimated signal model are given in Fig. 4.1(a). Also, the estimate of the driving frequency, the parameter estimates and the grid estimates are given in Figures 4.1(b)-4.1(f). Also, the adaptation of the gain and the prediction error are given in Fig. 4.2.

After 2000 samples the following estimates were obtained:

\[
\hat{X} = 1.4389, \quad \hat{\omega} = 0.3142, \quad \hat{f}_o = -0.0111, \\
\hat{g}_1 = \left( -1.6312 \quad -0.9377 \quad -0.4823 \quad 0.4929 \quad 0.9980 \quad 1.6546 \right)^T, \\
\hat{g}_2 = \left( -1.7056 \quad -1.0457 \quad -0.5968 \quad 0.4889 \quad 0.9552 \quad 1.6712 \right)^T, \\
\hat{\theta}_1 = \left( -2.1596 \quad -0.8547 \quad -0.3477 \quad 0.3052 \quad 0.8826 \quad 2.2133 \right)^T, \\
\hat{\theta}_2 = \left( -2.3649 \quad -0.9584 \quad -0.4104 \quad 0.3442 \quad 0.8855 \quad 2.2361 \right)^T.
\]

Also, the simulation was repeated under the same conditions using the fixed grid point algorithm introduced in Chapter 3. The true static nonlinearity, the estimated nonlinearity using the suggested grid point algorithm and the estimated nonlinearity using the algorithm of Chapter 3 are given in Fig. 4.3. As shown in Fig. 4.3, the adaptive grid point algorithm gives a better estimate for the static nonlinearity. Note that the compensated estimate for the nonlinearity is evaluated by multiplying the estimated nonlinearity by the \( \hat{X} \) to compensate for the nonlinearity static gain outside \( I_o. \) This gives a better view of the overall static gain of the model, which is the important point.

**Example 4.2. Comparison with the fixed grid point algorithm.**

In order to compare the performance of the adaptive grid point algorithm with the fixed grid point algorithm of Chapter 3, 100 Monte-Carlo simulations were performed with different noise realizations. The data were generated and the algorithms were initialized as in Example 4.1.

The mean square error (MSE) for the last 1000 samples of 8000 was evaluated for the two algorithms for different signal to noise ratios (SNRs). The results are plotted in Fig. 4.4, which show that the adaptive grid point algorithm gives lower MSE than the fixed grid point algorithm for moderate and high SNR. These results indicate that modeling errors are lower for the adaptive grid point algorithm. This is due to the fact that the adaptive grid point algorithm has more freedom to select suitable grid points.
4.4. Numerical Examples

(a) The signal (dashed) and the estimated signal model (solid)

(b) Convergence of the fundamental frequency

(c) Convergence of $\hat{\theta}_1$

(d) Convergence of $\hat{\theta}_2$

(e) Convergence of $\hat{g}_1$

(f) Convergence of $\hat{g}_2$

Figure 4.1: Convergence to the true parameter vector using Algorithm (4.8).
Figure 4.2: Convergence of $\hat{X}$ and the prediction error $\varepsilon$ for Example 4.1.

Figure 4.3: The compensated estimates of the static nonlinearity.

Figure 4.4: The MSE results.
Example 4.3. Tracking the fundamental frequency variations.

As mentioned in Chapter 3, to improve the ability of the suggested algorithm to track fundamental frequency variations, Algorithm (4.8) is modified to

\[
\varepsilon(t) = y(t) - \hat{y}(t)
\]

\[
S(t) = \psi^T(t)P(t-1)\psi(t) + r_2(t)
\]

\[
P(t) = P(t-1) - P(t-1)\psi(t)S^{-1}(t)\psi^T(t)P(t-1) + R_1(t)
\]

\[
\begin{bmatrix}
\hat{\theta}_1(t) \\
\hat{\theta}_2(t) \\
\hat{\theta}_3(t)
\end{bmatrix} = 
\begin{bmatrix}
\hat{\theta}_1(t-1) \\
\hat{\theta}_2(t-1) \\
\hat{\theta}_3(t-1)
\end{bmatrix} + P(t)\psi(t)\varepsilon(t) + D_M(4.14)
\]

where \(R_1(t)\) and \(r_2(t)\) are the gain design variables. The data were generated as in Example 4.1 with

\[
g_1 = \begin{bmatrix} -1 & -0.3 & -0.15 & 0.15 & 0.3 & 1 \end{bmatrix}^T,
\]

\[
g_2 = \begin{bmatrix} -1 & -0.3 & 0.3 & 1 \end{bmatrix}^T,
\]

\[
\theta_1 = \begin{bmatrix} -0.8 & -0.3 & 0.3 & 0.8 \end{bmatrix}^T, \quad u(t, \theta_1) \in I_1
\]

\[
\theta_2 = \begin{bmatrix} -0.8 & -0.5 & 0.5 & 0.8 \end{bmatrix}^T, \quad u(t, \theta_2) \in I_2.
\]

Also, Algorithm (4.14) was initialized as in Example 4.1. The design variables were \(R_1(t) = 10^{-6}\) except that \(R_1(2,2) = 5 \times 10^{-5}\) to speed up the frequency tracking and \(r_2(t) = 0.2\). Also, the grid points in (4.15) were used, and the initial values for the nonlinearities were given by straight lines with unity slope.

The signal and the estimated signal model are given in Fig. 4.5(a). Also, the true and estimated fundamental frequency are shown in Fig. 4.5(b). It can be concluded from the results that Algorithm (4.14) has the ability to track the fundamental frequency variations.
Example 4.4. Tracking the damped amplitude and the fundamental frequency variations.

In this example the driving wave of the system was given by 
\[ u(t, \theta_0^o) = e^{-t/\tau} \sin(\omega_0^o t) \]
where \( \omega_0^o = 2\pi \times 0.05 \) and \( \tau = 100 \). Two static nonlinearities were used;

\[
g_1^o = \begin{pmatrix} -1 & -0.3 & -0.15 & 0.15 & 0.3 & 1 \end{pmatrix}^T, \\
g_2^o = \begin{pmatrix} -1 & -0.3 & 0.3 & 1 \end{pmatrix}^T, \\
\theta_1^o = \begin{pmatrix} -0.8 & -0.3 & 0.3 & 0.8 \end{pmatrix}^T, \quad u(t, \theta_1^o) \in I_1 \\
\theta_2^o = \begin{pmatrix} -0.8 & -0.2 & 0.2 & 0.8 \end{pmatrix}^T, \quad u(t, \theta_2^o) \in I_2. 
\]

Algorithm (4.14) was initialized with \( P(0) = 0.01 I \), \( X = 1 \), \( K_0 = 1 \), \( f_o = 0 \) and \( \omega(0) = 2\pi \times 0.02 \). Furthermore, the design variables were \( r_2(t) = 0.1 \) and
4.4. Numerical Examples

Example 4.5. Signal modeling with different true parameter vector sizes.

In this example the data were generated and Algorithm (4.8) was initialized as in Example 4.1. The windowed periodogram with window width of 400 samples for the data and the model for two different numbers of the true parameters were also evaluated. The results are given in Fig. 4.7. The results show that the adaptive grid point algorithm gives good results for different parameter vector sizes.

Example 4.6. Performance of the adaptive grid point algorithm as compared to the CRB.

In order to compare the performance of Algorithm (4.8) with the derived CRB for the fundamental frequency estimation, 100 Monte-Carlo simulations of 2000 samples were performed with different noise realizations. The data were generated and the algorithm was initialized as in Example 4.3 except that \( P(0) = 10^{-4} I \). The statistics is based on excluding simulations that did not satisfy a margin of 5 standard deviations from the true fundamental frequency. Both the CRB for the fundamental frequency estimate and the mean square error (MSE) value were evaluated for different SNRs. The number of excluded simulations did not exceed 10 experiments. The results are plotted in Fig. 4.8, which show that the adaptive grid point algorithm gives good statistical results.
4.5 Conclusions

An adaptive grid point algorithm for periodic signal modeling has been presented. The suggested algorithm estimates the grid points in addition to the fundamental frequency and the parameters of the nonlinear output function of the signal model. Local convergence of the suggested algorithm was investigated by numerical examples and the CRB was calculated for this algorithm. Monte-Carlo experiments show that the suggested algorithm gives significantly better results than using the fixed grid point algorithm. Moreover, the algorithm can track both the amplitude and the fundamental frequency variations of a damped periodic signal.

4.A Proof of Theorem 4.1

Similarly to Appendix 3.C, the log-likelihood function is given by

\[
l(\theta) = \text{constant} - \frac{1}{2\sigma^2} \sum_{t=1}^{N} (y(t) - \hat{y}(t, \theta))^2, \tag{4.17}\]

Hence, the gradient of the log-likelihood function is defined as (cf. Eq. (4.4))

\[
\frac{\partial l(\theta)}{\partial \theta} = \left( \frac{\partial l(\theta)}{\partial \theta_1} \frac{\partial l(\theta)}{\partial \theta_n} \frac{\partial l(\theta)}{\partial \theta_g} \right), \tag{4.18}\]

where

\[
\frac{\partial l(\theta)}{\partial \theta_1} = \left( \frac{\partial l(\theta)}{\partial X} \frac{\partial l(\theta)}{\partial \omega} \right),
\]

\[
\frac{\partial l(\theta)}{\partial \theta_n} = \left( \frac{\partial l(\theta)}{\partial f_o} \frac{\partial l(\theta)}{\partial \theta_1} \cdots \frac{\partial l(\theta)}{\partial \theta_L} \right),
\]
In this case, the Fisher information matrix \[99\] can be written as

\[
\mathbf{J} = \mathbf{E} \left( \frac{\partial l(\theta)}{\partial \theta} \right)^T \left( \frac{\partial l(\theta)}{\partial \theta} \right),
\]

where

\[
\frac{\partial l(\theta)}{\partial \theta} = \left( \frac{\partial l(\theta)}{\partial f_{i-1}^j} \cdots \frac{\partial l(\theta)}{\partial f_{i+1}^j} \right),
\]

\[
\frac{\partial l(\theta)}{\partial g_{i-1}^j} = \left( \frac{\partial l(\theta)}{\partial g_{i-1}^j} \cdots \frac{\partial l(\theta)}{\partial g_{i+1}^j} \right),
\]

\[
\frac{\partial l(\theta)}{\partial u_{i-1}^j} = \left( \frac{\partial l(\theta)}{\partial u_{i-1}^j} \cdots \frac{\partial l(\theta)}{\partial u_{i+1}^j} \right).
\]

In order to calculate the blocks of the matrix \(\mathbf{J}\), the following relations are needed

\[
\frac{\partial l(\theta)}{\partial X} = \frac{1}{\sigma^2} \sum_{i=1}^{N} (y(t) - \hat{y}(t, \theta)) \frac{\partial \hat{y}(t, \theta)}{\partial X},
\]

\[
\frac{\partial l(\theta)}{\partial \omega} = \frac{1}{\sigma^2} \sum_{i=1}^{N} (y(t) - \hat{y}(t, \theta)) \frac{\partial \hat{y}(t, \theta)}{\partial \omega},
\]

\[
\frac{\partial l(\theta)}{\partial f_{i}^j} = \frac{1}{\sigma^2} \sum_{t=1}^{N} (y(t) - \hat{y}(t, \theta)) \frac{\partial f_{i}^j(\cdot)}{\partial f_{i}^j(\cdot)},
\]

\[
\frac{\partial l(\theta)}{\partial f_{i+1}^j} = \frac{1}{\sigma^2} \sum_{t=1}^{N} (y(t) - \hat{y}(t, \theta)) \frac{\partial f_{i+1}^j(\cdot)}{\partial f_{i+1}^j(\cdot)},
\]

\[
\frac{\partial l(\theta)}{\partial u_{i}^j} = \frac{1}{\sigma^2} \sum_{t=1}^{N} (y(t) - \hat{y}(t, \theta)) \frac{\partial f_{i}^j(\cdot)}{\partial u_{i}^j(\cdot)},
\]

\[
\frac{\partial l(\theta)}{\partial u_{i+1}^j} = \frac{1}{\sigma^2} \sum_{t=1}^{N} (y(t) - \hat{y}(t, \theta)) \frac{\partial f_{i+1}^j(\cdot)}{\partial u_{i+1}^j(\cdot)},
\]

where

\[
\frac{\partial \hat{y}(t, \theta)}{\partial X} = \left. \frac{\partial f_{i}^j(\cdot)}{\partial u} \right|_{\phi=\omega t},
\]

\[
\frac{\partial \hat{y}(t, \theta)}{\partial \omega} = \left. \frac{\partial f_{i}^j(\cdot)}{\partial u} \right|_{\phi=\omega t}.
\]
for $\tilde{u}(t, \theta_i) \in [u_{j-1}^i, u_j^i] \in I_j$, $i = -k^{-1}_j, \ldots, k^+_j - 1$, $j = 1, \ldots, L$. Thus using Assumption 3.1 and Assumption 3.11, it follows that

$$E \left[ \frac{\partial^2 l(\theta)}{\partial X^2} \right] = -\frac{1}{\sigma^2} \sum_{i=1}^{N} \left[ \frac{\partial f_j(\cdot)}{\partial u} \right] \Lambda^2(\phi) |_{\phi=\omega t},$$

$$E \left[ \frac{\partial^2 l(\theta)}{\partial \omega^2} \right] = -\frac{1}{\sigma^2} \sum_{i=1}^{N} \left[ \frac{\partial f_j(\cdot)}{\partial u} \right] X^2 \lambda^2 \left[ \frac{d\Lambda(\phi)}{d\phi} \right] |_{\phi=\omega t}^2,$$

$$E \left[ \frac{\partial^2 l(\theta)}{\partial f_o^2} \right] = -\frac{1}{\sigma^2} \sum_{i=1}^{N} \left[ \frac{\partial f_j(\cdot)}{\partial f_o} \right]^2,$$

$$E \left[ \frac{\partial^2 l(\theta)}{\partial X \partial f_o^{t+1}} \right] = -\frac{1}{\sigma^2} \sum_{i=1}^{N} \left[ \frac{\partial f_j(\cdot)}{\partial f_o} \right] X \lambda \left[ \frac{d\Lambda(\phi)}{d\phi} \right] |_{\phi=\omega t},$$

$$E \left[ \frac{\partial^2 l(\theta)}{\partial X \partial f_i^{t+1}} \right] = -\frac{1}{\sigma^2} \sum_{i=1}^{N} \left[ \frac{\partial f_j(\cdot)}{\partial f_i} \right] \lambda(\phi) |_{\phi=\omega t} + F,$$

$$E \left[ \frac{\partial^2 l(\theta)}{\partial X \partial u^{t+1}_j} \right] = -\frac{1}{\sigma^2} \sum_{i=1}^{N} \left[ \frac{\partial f_j(\cdot)}{\partial u^{t+1}} \right] \lambda(\phi) |_{\phi=\omega t},$$

$$E \left[ \frac{\partial^2 l(\theta)}{\partial \omega \partial f_i^{t+1}} \right] = -\frac{1}{\sigma^2} \sum_{i=1}^{N} \left[ \frac{\partial f_j(\cdot)}{\partial f_i} \right] X \lambda \left[ \frac{d\Lambda(\phi)}{d\phi} \right] |_{\phi=\omega t}$$

$$E \left[ \frac{\partial^2 l(\theta)}{\partial \omega \partial u^{t+1}_i} \right] = -\frac{1}{\sigma^2} \sum_{i=1}^{N} \left[ \frac{\partial f_j(\cdot)}{\partial u^{t+1}} \right] \lambda(\phi) |_{\phi=\omega t},$$

$$E \left[ \frac{\partial^2 l(\theta)}{\partial f_o \partial f_i^{t+1}} \right] = -\frac{1}{\sigma^2} \sum_{i=1}^{N} \left[ \frac{\partial f_j(\cdot)}{\partial f_o} \right] \lambda(\phi) |_{\phi=\omega t}, \quad (4.23)$$
Taking into account that \( \frac{\partial^2 l(\theta)}{\partial \theta_m \partial \theta_n} = 0 \) and \( \frac{\partial^2 l(\theta)}{\partial \theta_m \partial \theta_n} = 0 \) for \( m \neq n \). Introducing the notation in (4.11) and noticing that \( J = 1/\sigma^2 \sum_{t=1}^{N} I(t) \), Eq. (4.9) directly follows from Assumption 3.12.

**Remark 4.2.** \( \frac{\partial f_j(\cdot)}{\partial u} \), \( \frac{\partial f_j(\cdot)}{\partial f_i} \), \( \frac{\partial f_j(\cdot)}{\partial f_i} \), \( \frac{\partial f_j(\cdot)}{\partial u_{i+1}} \) and \( \frac{\partial f_j(\cdot)}{\partial u_{i+1}} \) can be calculated for different subintervals using (4.8). Thus the matrix \( I(t) \) takes the following forms:
(i) When \( \hat{u}(t, \theta_i) \in [u_i^j, u_{i+1}^j] \) for all \( i \) of the form
\[
\begin{cases}
-k^i_1, \ldots, -2, 1, \ldots, k^i_1 - 1, & j = 1, \\
-k^i_j, \ldots, k^i_j - 1, & j = 2, \ldots, L.
\end{cases}
\]

\[
I(t) = \begin{pmatrix}
I_{X,X} & I_{X,\omega} & 0 & \cdots & 0 & I_{X,\omega_{i+1}} & 0 & \cdots & 0 \\
I_{\omega,X} & I_{\omega,\omega} & 0 & \cdots & 0 & I_{\omega,\omega_{i+1}} & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
I_{\omega_{i+1},X} & I_{\omega_{i+1},\omega} & 0 & \cdots & 0 & I_{\omega_{i+1},\omega_{i+1}} & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0
\end{pmatrix},
\]

where
\[
I_{X,X} = \left[ \frac{f^j_{i+1} - f^j_i}{u_{i+1}^j - u_i^j} \right]^2 \Lambda^2(\phi) \mid_{\phi=\omega t},
\]
\[
I_{\omega,\omega} = \left[ \frac{f^j_{i+1} - f^j_i}{u_{i+1}^j - u_i^j} \right]^2 \chi^2 X t^2 \left[ \frac{d\Lambda(\phi)}{d\phi} \right]^2 \mid_{\phi=\omega t},
\]
\[
I_{X,\omega} = \left[ \frac{f^j_{i+1} - f^j_i}{u_{i+1}^j - u_i^j} \right]^2 X t \Lambda(\phi) \frac{d\Lambda(\phi)}{d\phi} \mid_{\phi=\omega t},
\]
\[
I_{X,f^j_{i+1}} = \left[ \frac{f^j_{i+1} - f^j_i}{u_{i+1}^j - u_i^j} \right] \frac{u_{i+1}^j - u(t, \theta_i)}{u_{i+1}^j - u_i^j} \Lambda(\phi) \mid_{\phi=\omega t},
\]
\[
I_{X,f^j_{i+1}} = \left[ \frac{f^j_{i+1} - f^j_i}{u_{i+1}^j - u_i^j} \right] \frac{u(t, \theta_i) - u_i^j}{u_{i+1}^j - u_i^j} \Lambda(\phi) \mid_{\phi=\omega t},
\]
\[
I_{X,u_{i+1}^j} = \left[ \frac{f^j_{i+1} - f^j_i}{u_{i+1}^j - u_i^j} \right] \left[ (f^j_{i+1} - f^j_i) u_{i+1}^j + \frac{f^j_{i+1} - f^j_i}{(u_{i+1}^j - u_i^j)^2} u(t, \theta_i) \right] \Lambda(\phi) \mid_{\phi=\omega t},
\]
\[
I_{X,u_{i+1}^j} = \left[ \frac{f^j_{i+1} - f^j_i}{u_{i+1}^j - u_i^j} \right] \left[ (f^j_{i+1} - f^j_i) u_i^j + \frac{f^j_{i+1} - f^j_i}{(u_{i+1}^j - u_i^j)^2} u(t, \theta_i) \right] \Lambda(\phi) \mid_{\phi=\omega t},
\]
\[
I_{\omega,f^j_{i+1}} = \left[ \frac{f^j_{i+1} - f^j_i}{u_{i+1}^j - u_i^j} \right] \frac{u_{i+1}^j - u(t, \theta_i)}{u_{i+1}^j - u_i^j} X t \frac{d\Lambda(\phi)}{d\phi} \mid_{\phi=\omega t},
\]
\[
I_{\omega,f^j_{i+1}} = \left[ \frac{f^j_{i+1} - f^j_i}{u_{i+1}^j - u_i^j} \right] \frac{u(t, \theta_i) - u_i^j}{u_{i+1}^j - u_i^j} X t \frac{d\Lambda(\phi)}{d\phi} \mid_{\phi=\omega t}.
\]
\[ I_{\omega, u_i} = \frac{f^1_{i+1} - f^1_i}{u^1_{i+1} - u^1_i} \left[ (f^1_i - f^1_{i+1})u^1_{i+1} + \frac{f^3_{i+1} - f^3_i}{(u^1_{i+1} - u^1_i)^2} u(t, \theta_i) \right] X_t \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi = \omega t} \]

\[ I_{\omega, u^1_{i+1}} = \frac{f^3_{i+1} - f^3_i}{u^2_{i+1} - u^2_i} \left[ (f^3_i - f^3_{i+1})u^2_{i+1} - \frac{f^3_{i+1} - f^3_i}{(u^2_{i+1} - u^2_i)^2} u(t, \theta_i) \right] X_t \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi = \omega t} \]

\[ I_{f^1_i, f^1_i} = \left[ u^1_{i+1} - u(t, \theta_i) \right]^2 \]

\[ I_{f^1_{i+1}, f^1_{i+1}} = \frac{u^1_{i+1} - u(t, \theta_i)}{u^1_{i+1} - u^1_i} u(t, \theta_i) - u^1_{i+1}, \]

\[ I_{u^1_i, u^1_i} = \left[ \frac{(f^1_i - f^1_{i+1})u^1_{i+1} + \frac{f^3_{i+1} - f^3_i}{(u^1_{i+1} - u^1_i)^2} u(t, \theta_i)}{(u^1_{i+1} - u^1_i)^2} \right]^2, \]

\[ I_{u^1_{i+1}, u^1_{i+1}} = \left[ \frac{(f^3_i - f^3_{i+1})u^1_{i+1} + \frac{f^3_{i+1} - f^3_i}{(u^1_{i+1} - u^1_i)^2} u(t, \theta_i)}{(u^1_{i+1} - u^1_i)^2} \right]^2, \]

\[ I_{u^1_i, u^1_{i+1}} = \left[ \frac{(f^1_i - f^1_{i+1})u^1_{i+1} + \frac{f^3_{i+1} - f^3_i}{(u^1_{i+1} - u^1_i)^2} u(t, \theta_i)}{(u^1_{i+1} - u^1_i)^2} \right]^2 \]

\[ (f^1_i - f^1_{i+1})u^1_{i+1} + \frac{f^3_{i+1} - f^3_i}{(u^1_{i+1} - u^1_i)^2} u(t, \theta_i) \]

\[ I_{f^1_i, u^1_i} = \left[ u^1_{i+1} - u(t, \theta_i) \right]^2 \]

\[ I_{f^1_{i+1}, u^1_i} = \left[ \frac{(f^1_i - f^1_{i+1})u^1_{i+1} + \frac{f^3_{i+1} - f^3_i}{(u^1_{i+1} - u^1_i)^2} u(t, \theta_i)}{(u^1_{i+1} - u^1_i)^2} \right]^2, \]

\[ I_{f^1_{i+1}, u^1_{i+1}} = \left[ \frac{(f^3_i - f^3_{i+1})u^1_{i+1} + \frac{f^3_{i+1} - f^3_i}{(u^1_{i+1} - u^1_i)^2} u(t, \theta_i)}{(u^1_{i+1} - u^1_i)^2} \right]^2, \]

\[ I_{f^1_i, u^1_{i+1}} = \left[ \frac{(f^1_i - f^1_{i+1})u^1_{i+1} + \frac{f^3_{i+1} - f^3_i}{(u^1_{i+1} - u^1_i)^2} u(t, \theta_i)}{(u^1_{i+1} - u^1_i)^2} \right]^2 \]
(ii) When \( \hat{u}(t, \theta) \in [u_{-1}, u_0] \subset I_1; \)

\[
I(t) = \begin{pmatrix}
I_{X,X} & I_{X,\omega} & I_{X,f_0} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
I_{\omega,X} & I_{\omega,\omega} & I_{\omega,f_0} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
I_{f_0,X} & I_{f_0,\omega} & I_{f_0,f_0} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},
\]

where

\[
I_{X,X} = \left[ \frac{K_o u_o - f_o - f_1}{u_o - u_{-1}} \right]^2 \lambda^2(\phi) \bigg|_{\phi=\omega t},
\]

\[
I_{\omega,\omega} = \left[ \frac{K_o u_o - f_o - f_1}{u_o - u_{-1}} \right]^2 \lambda^2 t^2 \left[ \frac{d\lambda(\phi)}{d\phi} \right] \bigg|_{\phi=\omega t},
\]

\[
I_{f_0,f_0} = \left[ \frac{u(t, \theta) - u_{-1}}{u_o - u_{-1}} \right]^2,
\]

\[
I_{X,\omega} = \left[ \frac{K_o u_o - f_o - f_1}{u_o - u_{-1}} \right]^2 \lambda \left( I_1 \right) \left( \frac{d\lambda(\phi)}{d\phi} \right) \bigg|_{\phi=\omega t},
\]

\[
I_{X,f_0} = \left[ \frac{K_o u_o - f_o - f_1}{u_o - u_{-1}} \right] \frac{u(t, \theta) - u_{-1}}{u_o - u_{-1}} \lambda \left( I_1 \right) \left( \frac{d\lambda(\phi)}{d\phi} \right) \bigg|_{\phi=\omega t},
\]

\[
I_{\omega,f_0} = \left[ \frac{K_o u_o - f_o - f_1}{u_o - u_{-1}} \right] \frac{u(t, \theta) - u_{-1}}{u_o - u_{-1}} \lambda \left( I_1 \right) \left( \frac{d\lambda(\phi)}{d\phi} \right) \bigg|_{\phi=\omega t},
\]

\[
I_{X,u_{-1}} = \left[ \frac{K_o u_o - f_o - f_1}{u_o - u_{-1}} \right] \times
\]

\[
\left[ \frac{(f_1 - K_o u_o - f_o) u_o}{(u_o - u_{-1})^2} + \frac{K_o u_o - f_o - f_1}{(u_o - u_{-1})^2} u(t, \theta) \right] \lambda(\phi) \bigg|_{\phi=\omega t},
\]

\[
I_{\omega,u_{-1}} = \left[ \frac{K_o u_o - f_o - f_1}{u_o - u_{-1}} \right] \times
\]

\[
\left[ \frac{(f_1 - K_o u_o - f_o) u_o}{(u_o - u_{-1})^2} + \frac{K_o u_o - f_o - f_1}{(u_o - u_{-1})^2} u(t, \theta) \right] \lambda \left( I_1 \right) \left( \frac{d\lambda(\phi)}{d\phi} \right) \bigg|_{\phi=\omega t},
\]

(4.26)
4.A. Proof of Theorem 4.1

\[ I_{f_{o}, f_{u_{-1}}} = \frac{u(t, \theta_{i}) - u_{-1}}{u_{o_{-1}} - u_{-1}} \frac{u_{o_{-}} - u(t, \theta_{i})}{u_{o_{-}} - u_{-1}}, \]

\[ I_{f_{o}, u_{-1}} = \frac{u(t, \theta_{i}) - u_{-1}}{u_{o_{-1}} - u_{-1}} \left[ \frac{(f_{-1} - K_{a_{o_{-}}} - f_{a})u_{o_{-}}}{(u_{o_{-}} - u_{-1})^2} + \frac{K_{a_{o_{-}}} + f_{a} - f_{-1}}{(u_{o_{-}} - u_{-1})^2} u(t, \theta_{i}) \right], \]

\[ I_{f_{1}, f_{-1}} = \left[ \frac{u_{o_{-}} - u(t, \theta_{i})}{u_{o_{-}} - u_{-1}} \right]^2, \]

\[ I_{u_{-1}, u_{-1}} = \left[ \frac{(f_{-1} - K_{a_{o_{-}}} - f_{a})u_{o_{-}}}{(u_{o_{-}} - u_{-1})^2} + \frac{K_{a_{o_{-}}} + f_{a} - f_{-1}}{(u_{o_{-}} - u_{-1})^2} u(t, \theta_{i}) \right]^2, \]

\[ I_{f_{1}, u_{-1}} = \left[ \frac{u_{o_{-}} - u(t, \theta_{i})}{u_{o_{-}} - u_{-1}} \right] \left[ \frac{(f_{-1} - K_{a_{o_{-}}} - f_{a})u_{o_{-}}}{(u_{o_{-}} - u_{-1})^2} + \frac{K_{a_{o_{-}}} + f_{a} - f_{-1}}{(u_{o_{-}} - u_{-1})^2} u(t, \theta_{i}) \right]. \]

(iii) When \( \tilde{u}(t, \theta_{i}) \in [u_{o_{-}}, u_{o_{+}}] \subset I_{1}; \)

\[ I(t) = \begin{pmatrix}
I_{x,x} & I_{x,w} & I_{x,f_{o}} & 0 & \cdots & 0
I_{w,x} & I_{w,w} & I_{w,f_{o}} & 0 & \cdots & 0
I_{f_{o}, f_{u}} & I_{f_{o}, w} & I_{f_{o}, f_{o}} & 0 & \cdots & 0
0 & 0 & 0 & 0 & \cdots & 0
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots
\end{pmatrix}, \]  

(4.28)

where

\[ I_{x,x} = K_{a_{o}}^2 \Lambda^2(\phi) \bigg|_{\phi = \omega t}, \]

\[ I_{x,w} = K_{a_{o}}^2 X^2 \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi = \omega t}, \]

\[ I_{f_{o}, f_{o}} = 1, \]

\[ I_{x,f_{o}} = K_{a_{o}} X t \Lambda(\phi) \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi = \omega t}, \]

\[ I_{w,f_{o}} = K_{a_{o}} X t \frac{d\Lambda(\phi)}{d\phi} \bigg|_{\phi = \omega t}. \]

(iv) When \( \tilde{u}(t, \theta_{i}) \in [u_{o_{+}}, u_{1}] \subset I_{1}; \)

\[ I(t) = \begin{pmatrix}
I_{x,x} & I_{x,w} & I_{x,f_{o}} & 0 & \cdots & 0
I_{w,x} & I_{w,w} & I_{w,f_{o}} & 0 & \cdots & 0
I_{f_{o}, x} & I_{f_{o}, w} & I_{f_{o}, f_{o}} & 0 & \cdots & 0
0 & 0 & 0 & 0 & \cdots & 0
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots
\end{pmatrix}, \]  

(4.30)
\[ I_{X,X} = \left[ \frac{f_1 - K_o u_{o+} - f_o}{u_1 - u_{o+}} \right]^2 \Lambda^2(\phi) \big|_{\phi=\omega t}, \]

\[ I_{\omega,\omega} = \left[ \frac{f_1 - K_o u_{o+} - f_o}{u_1 - u_{o+}} \right]^2 X^2 t \frac{d\Lambda(\phi)}{d\phi} \big|_{\phi=\omega t}. \]

\[ I_{f_o,fo} = \left[ \frac{u_1 - u(t,\theta)}{u_1 - u_{o+}} \right]^2, \]

\[ I_{X,\omega} = \left[ \frac{f_1 - K_o u_{o+} - f_o}{u_1 - u_{o+}} \right] X t \Lambda(\phi) \frac{d\Lambda(\phi)}{d\phi} \big|_{\phi=\omega t}, \]

\[ I_{X,f_o} = \frac{f_1 - K_o u_{o+} - f_o}{u_1 - u_{o+}} \left( u_1 - u(t,\theta) \right) \frac{\Lambda(\phi)}{\phi=\omega t}, \]

\[ I_{X,\omega} = \frac{f_1 - K_o u_{o+} - f_o}{u_1 - u_{o+}} \left( u_1 - u(t,\theta) \right) \frac{\Lambda(\phi)}{\phi=\omega t}, \]

\[ I_{x,f_o} = \frac{f_1 - K_o u_{o+} - f_o}{u_1 - u_{o+}} \left( u_1 - u(t,\theta) \right) \frac{\Lambda(\phi)}{\phi=\omega t}, \]

\[ I_{X,u_1} = \frac{f_1 - K_o u_{o+} - f_o}{u_1 - u_{o+}} \times \]

\[ \left( \frac{(f_1 - K_o u_{o+} - f_o)u_{o+}}{(u_1 - u_{o+})^2} - \frac{f_1 - K_o u_{o+} - f_o}{u_1 - u_{o+}} \right) \frac{\Lambda(\phi)}{\phi=\omega t}, \]

\[ I_{\omega,u_1} = \frac{f_1 - K_o u_{o+} - f_o}{u_1 - u_{o+}} \times \]

\[ \left( \frac{(f_1 - K_o u_{o+} - f_o)u_{o+}}{(u_1 - u_{o+})^2} - \frac{f_1 - K_o u_{o+} - f_o}{u_1 - u_{o+}} \right) \frac{\Lambda(\phi)}{\phi=\omega t}, \]

\[ I_{f_o,f_1} = \frac{u_1 - u(t,\theta)}{u_1 - u_{o+}} \left( u(t,\theta) - u_{o+} \right), \]

\[ I_{f_o,u_1} = \frac{u_1 - u(t,\theta)}{u_1 - u_{o+}} \left( \frac{(f_1 - K_o u_{o+} - f_o)u_{o+}}{(u_1 - u_{o+})^2} - \frac{f_1 - K_o u_{o+} - f_o}{u_1 - u_{o+}} \right) \left( u(t,\theta) \right), \]

\[ I_{f_1,f_1} = \left( \frac{u(t,\theta) - u_{o+}}{u_1 - u_{o+}} \right)^2, \]

\[ I_{u_1,u_1} = \frac{\left( \frac{(f_1 - K_o u_{o+} - f_o)u_{o+}}{(u_1 - u_{o+})^2} - \frac{f_1 - K_o u_{o+} - f_o}{u_1 - u_{o+}} \right)^2}{(u_1 - u_{o+})^2}, \]

\[ I_{f_1,u_1} = \frac{\left( \frac{(f_1 - K_o u_{o+} - f_o)u_{o+}}{(u_1 - u_{o+})^2} - \frac{f_1 - K_o u_{o+} - f_o}{u_1 - u_{o+}} \right)^2}{(u_1 - u_{o+})^2}. \]
Part II

Periodic Signal Modeling Using Orbits of Second-Order Nonlinear ODE's
Chapter 5

The Second-Order Nonlinear ODE Model

5.1 Introduction

The work of this part of the thesis is inspired by one possible model for the generation of periodic signals, namely nonlinear ordinary differential equations (ODE’s). There is a rich theory on the subject as outlined in, e.g., [62, 84, 92, 104, 105, 111, 125, 128]. The focus here will be on periodic orbits and their properties. Some of the strongest results of the theory concern ODE’s with two state variables. The reason is that closed periodic orbits, in case they do not intersect themselves, divide the space into one part interior to the orbit and one part exterior to the orbit. The mathematical consequence is that there are several powerful theorems on the existence of periodic solutions to ODE’s, and hence it seems to be advantageous to base estimation algorithms on second-order ODE’s.

Many systems that generate periodic signals can be described by nonlinear ODE’s. Examples include tunnel diodes, pendulums, biological predator-prey systems and radio frequency synthesizers, see [62]. Many of these systems are described by second-order ODE’s with polynomial right hand sides. It can therefore be expected that there are good opportunities to obtain highly accurate models by estimating only a few parameters. The parsimony principle, see e.g. [99], suggests that the achievable accuracy would be improved by the proposed methods, as compared, e.g., to the periodogram and other methods that do not impose the same amount of prior information on the solution.

The specific signal model of this part is therefore obtained by introducing a polynomial parameterization of the right hand side of a general second-order nonlinear ODE, and by defining the periodic signal to be modeled as a function of the states of this ODE.

The chapter is organized as follows. Some motivation examples are given in Section 5.2. Section 5.3 introduces the the assumptions on the modeled signal and the measurement noise. Section 5.4 discusses different model structures. Section 5.5 presents the polynomial parameterization of the ODE model. Discretization of the parameterized ODE model is introduced in Section 5.6.
5.2 Examples

In this section two examples for oscillating systems that can be described by second-order nonlinear ODE’s are given. The examples are taken from [62] with some changes.

Example 5.1. The simple pendulum.
Consider the simple pendulum shown in Fig. 5.1. The simple pendulum consists of a mass \( \bar{m} \) hanging from a string of length \( \lambda \) and fixed at a pivot point. When displaced to an initial angle and released, the pendulum will swing back and forth with periodic motion. By applying Newton’s second law of motion, the equation of motion in the tangential direction can be written as

\[
\ddot{\theta} \bar{m} \lambda = -\bar{m} g \sin \theta - \xi \lambda \dot{\theta},
\]

(5.1)

where \( g \) is the acceleration due to gravity and \( \xi \) is the coefficient of friction.

Choosing the state variables as

\[
\begin{pmatrix}
  x_1 \\
  x_2
\end{pmatrix} = \begin{pmatrix}
  \theta \\
  \dot{\theta}
\end{pmatrix},
\]

(5.2)

the state equations of the simple pendulum are

\[
\begin{aligned}
  x_1 &= x_2, \\
  x_2 &= -\frac{g}{\lambda} \sin x_1 - \frac{\xi}{\bar{m}} x_2.
\end{aligned}
\]

(5.3)

In case of no friction, the state equations become

\[
\begin{aligned}
  x_1 &= x_2, \\
  x_2 &= -\frac{g}{\lambda} \sin x_1.
\end{aligned}
\]

(5.4)

These two state equations were numerically solved for \( g = 9.81 \text{ m/s}^2 \) and \( \lambda = 1 \text{ m} \). The sampling period was selected as \( h = 0.01 \text{ s} \). The pendulum was
initiated at rest with a 90 degree angle with the vertical. The results are plotted in Fig. 5.2. Figure 5.2(a) shows that the angle $\theta$ is oscillating with time and Fig. 5.2(b) shows that plotting $x_2 = \dot{\theta}$ vs. $x_1 = \theta$ represents a periodic orbit.

**Example 5.2. The negative resistance oscillator.**

The negative resistance oscillator of Fig. (5.3) consists of an inductor $L$, a capacitor $C$ and a resistive element. The resistive element has the following $i$-$v$ characteristic:

\[ i = h(v), \]  

(5.5)
where \( h(\cdot) \) satisfies the following conditions:
\[
\begin{align*}
  h(0) &= 0, & \frac{dh}{dv} \bigg|_{v=0} &< 0, \\
  h(v) &\to \infty \text{ as } v \to \infty, \\
  h(v) &\to -\infty \text{ as } v \to -\infty.
\end{align*}
\]

Applying Kirchhoff’s current law at node \( n \)
\[
i_C + i_L + i = 0. \tag{5.6}
\]
Substituting by Eq. (5.5) and expressing the currents \( i_C \) and \( i_L \) as a function of the voltage \( v \), Eq. (5.6) becomes
\[
C \frac{dv}{dt} + \frac{1}{L} \int_{-\infty}^{t} v(t) dt + h(v) = 0. \tag{5.7}
\]
Differentiating Eq. (5.7) with respect to time \( t \) and multiplying the result by \( L \), results in
\[
CL \frac{d^2v}{dt^2} + v + L \frac{dh(v)}{dv} = 0. \tag{5.8}
\]
Now, changing the time variable from \( t \) to \( \tau = \frac{t}{\sqrt{CL}} \), transforms Eq. (5.8) to
\[
\ddot{v} + \varepsilon \frac{dh(v)}{dv} \dot{v} + v = 0, \tag{5.9}
\]
where \( \dot{v} \) and \( \ddot{v} \) denote the first and second derivatives with respect to \( \tau \), respectively, and \( \varepsilon = \sqrt{L/C} \).

Choosing the state variables as
\[
\begin{pmatrix}
x_1 \\
x_2
\end{pmatrix} = \begin{pmatrix}
v \\
\dot{v}
\end{pmatrix}, \tag{5.10}
\]

Figure 5.4: Oscillations of the negative resistance oscillator.
the state equations representing the negative resistance oscillator become

\[
\begin{align*}
    x_1 &= x_2, \\
    x_2 &= -x_1 - \varepsilon \frac{dh(x_1)}{dx_1} x_2.
\end{align*}
\] (5.11)

These two state equations were numerically solved for \( h(x_1) = -x_1 + \frac{1}{3} x_1^3 \) (the Van der Pol oscillator) and \( \varepsilon = 1 \). The results are given in Fig. (5.4).

The previous two examples show some practical oscillating systems that can be described by second-order nonlinear ODE’s. In case it is required to model the oscillations generated by these systems, a second-order nonlinear ODE model is a good match. Hence, an estimation technique for modeling or identifying the right hand sides of Eq. (5.4) and Eq. (5.11) is needed. In this case, the accuracy of the obtained ODE model is expected to exceed the accuracy of other models that do not make use of such priors. In the rest of this chapter, the second-order ODE model suggested for modeling periodic signals as being oscillations such as the ones given in Fig. 5.2 and Fig. 5.4 is described.

5.3 Modeled Signal and Measurements

The starting point is the discrete time measured signal \( z(kh) \), where

\[
    z(kh) = y(kh) + e(kh), \quad k = 1, \cdots, N.
\] (5.12)

Here

- \( y(t) \) is the continuous time signal to be modeled;
- \( h \) is the sampling interval;
- \( y(kh) \) is the sampled value of \( y(t) \);
- \( e(kh) \) is the discrete time measurement noise.

It is assumed that \( y(t) \) is periodic, \textit{i.e.}

\textbf{Assumption 5.1.} \( y(t + T) = y(t), \forall t \in R, 0 < T < \infty \), where \( T \) denotes the period.

Furthermore, \( e(kh) \) is assumed to be zero-mean Gaussian white noise, \textit{i.e.}

\textbf{Assumption 5.2.} \( e(kh) \in \mathcal{N}(0, \sigma^2), \ E[e(kh)e(kh + jh)] = \delta_{j,0}\sigma^2. \)

5.4 Model Structures

As stated above, the main idea of the work in Part II of this thesis is to model the generation of the signal \( y(t) \) by means of an unknown parameter vector \( \theta \)
and a nonlinear ODE, i.e.
\[
\begin{align*}
\dot{x} &= f(x, \theta), \\
y &= h(x).
\end{align*}
\] (5.13)

Considering the use of second-order ODE’s, Eq. (5.13) is modified to
\[
\begin{align*}
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{pmatrix} &= \begin{pmatrix}
f_1(x_1(t), x_2(t), \theta_1) \\
f_2(x_1(t), x_2(t), \theta_2)
\end{pmatrix}, \\
y(t) &= \begin{pmatrix} c_1 & c_2 \end{pmatrix} \begin{pmatrix} x_1(t) \\
x_2(t) \end{pmatrix}.
\end{align*}
\] (5.14)

Here
\[
\theta = \begin{pmatrix} \theta^T_1 & \theta^T_2 \end{pmatrix}^T,
\] (5.15)
where the vectors \( \theta_1 \) and \( \theta_2 \) are unknown parameter vectors and \( \begin{pmatrix} c_1 & c_2 \end{pmatrix} \) contains the selected output weighting factors.

At this point it is highly relevant to pose the question whether the model (5.14) may be too general. In [122, 123], it is proved that it can often be assumed that the second order ODE
\[
\ddot{y}(t) = f(y(t), \dot{y}(t), \theta),
\] (5.16)
genrates the periodic signal that is measured provided that the following condition holds:

**Assumption 5.3.** The phase plane plot that is constructed from the periodic signal \( y(t) \) and its first derivative \( \dot{y}(t) \) lacks any intersections or limiting cases such as corners, stops and cusps.

**Remark 5.1.** Assumption 5.3 is needed to exclude signal classes that have intersected phase plane plots since these classes need higher order ODE’s to be modeled accurately, see Chapter 6 and [122, 123] for more details.

Thus choosing the state variables as follows
\[
\begin{pmatrix} x_1 \\
x_2 \end{pmatrix} = \begin{pmatrix} y(t) \\
\dot{y}(t) \end{pmatrix},
\] (5.17)
the model given in (5.14) is reduced to
\[
\begin{align*}
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{pmatrix} &= \begin{pmatrix} x_2(t) \\
f(x_1(t), x_2(t), \theta) \end{pmatrix}, \\
y(t) &= \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\
x_2(t) \end{pmatrix}.
\end{align*}
\] (5.18)

This model depends only on the parameters of the right hand side function of the second state equation of (5.18), a fact that should be advantageous from a computational and performance point of view. The obvious question is then - are there other facts that could motivate the use of (5.14) instead of (5.18)?
One key to the answer of this question is the output equations of (5.14) and (5.18). Physically, it may very well be the case that the system is governed by (5.16), but the generated periodic signal may be a nonlinear function of $y$ and $\dot{y}$. It could also be the case that this function is not well known and/or noninvertible, and the question is then what can be done in such situations? One idea would be to reuse the output relation of (5.18) and try to find a representation of the state space ODE such that the first state equals the sought output. In order to study the implications of this idea, the following system is studied

$$\ddot{y}(t) = f(y(t), \dot{y}(t), \theta), \quad (5.19)$$
$$z = g(y(t), \dot{y}(t)). \quad (5.20)$$

With the states chosen as

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} z \\ \dot{y}(t) \end{pmatrix}, \quad (5.21)$$

it follows by differentiation that

$$\dot{x}_1 = \frac{\partial g(y, x_2)}{\partial y} x_2 + \frac{\partial g(y, x_2)}{\partial x_2} f(y(t), x_2, \theta), \quad (5.22)$$
$$\dot{x}_2 = f(y(t), x_2, \theta). \quad (5.23)$$

It remains to find an expression for $y$ as a function of $x_1$ and $x_2$. This follows from Eqs. (5.19)-(5.21), since under mild regularity conditions [62] the inverse function theorem allows at least a local solution of the output equation with respect to $y$, i.e.

$$x_1 = g(y, x_2) \Rightarrow y = h(x_1, x_2). \quad (5.24)$$

When (5.24) is inserted in (5.22) and (5.23), it is clear that the result fits into the structure (5.14), with the weight of the output equation being equal to those of (5.18). The conclusion is that the model structure (5.14) may extend the validity of the proposed periodic signal modeling approach, beyond that of (5.18). Despite this fact, the study in this thesis will concentrate on the model structure (5.18) since the analysis of Chapter 6 proves that (5.18) is sufficient to model periodic signals that satisfy Assumption 5.3.

### 5.5 Parameterization

A natural approach is now to expand the right hand side of the second state equation of (5.18) in terms of known basis functions \( \{b_k(x_1(t), x_2(t))\}_{k=0}^{\infty} \), modeling the right hand side as a truncated superposition of these functions. In case of a polynomial model, a suitable parameterization is

$$f(x_1(t), x_2(t), \theta) = \sum_{l=0}^{L} \sum_{m=0}^{M} \theta_{l,m} x_1^l(t)x_2^m(t), \quad (5.25)$$

$$\theta = ( \theta_{0,0} \cdots \theta_{0,M} \cdots \theta_{L,0} \cdots \theta_{L,M} )^T. \quad (5.26)$$
Remark 5.2. No scale factor problems are expected with this parameterization. The reason is that the transformation
\[ \bar{y} = ky, \] transforms (5.16) to
\[ \ddot{\bar{y}}(t) = kf \left( \frac{1}{k} \bar{y}(t), \frac{1}{k} \dot{\bar{y}}(t), \theta \right). \] Since the polynomial model is a general function expansion, the function of (5.28) can be modeled equally well as the function of (5.16).

5.6 Discretization

In order to formulate complete discrete time models, the continuous time ODE model (5.18) needs to be discretized. This can be done, for example, by exploiting Euler numerical integration schemes. The discretization interval is selected to be equal to the sampling period \( h \). The numerical integration schemes considered in this thesis and its corresponding discrete models are:

- Euler backward approximation (EB):
  \[ x_1(kh) = x_1(kh-h) + hx_2(kh), \] \[ x_2(kh) = x_2(kh-h) + \sum_{l=0}^{L} \sum_{m=0}^{M} \theta_{l,m} x_1^l(kh)x_2^m(kh). \] (5.30)

- Euler forward approximation (EF):
  \[ x_1(kh+h) = x_1(kh) + hx_2(kh), \] \[ x_2(kh+h) = x_2(kh) + \sum_{l=0}^{L} \sum_{m=0}^{M} \theta_{l,m} x_1^l(kh)x_2^m(kh). \] (5.32)

- Euler center approximation (EC):
  \[ x_1(kh+h) = x_1(kh-h) + 2hx_2(kh), \] \[ x_2(kh+h) = x_2(kh-h) + 2h \sum_{l=0}^{L} \sum_{m=0}^{M} \theta_{l,m} x_1^l(kh)x_2^m(kh). \] (5.34)

Remark 5.3. For simplicity the discretization interval is selected to be equal to the sampling interval \( h \). Otherwise, it would be a multiple of \( h \).

Remark 5.4. Using (5.30), in which \( x_2(kh) = g(x_1(kh), x_2(kh)) \), does not lead to any numerical problems since (5.30) will not be solved for \( x_2(kh) \). As will be shown in Chapter 7, \( x_1(kh) \) and \( x_2(kh) \) will be replaced by quantities evaluated from the measured data.

Now the discretized model can be used to implement different off-line and on-line algorithms to estimate the parameter vector \( \theta \). This is the main theme of Chapters 7-12. In the next chapter, the sufficiency of the second-order ODE model described by Eq. (5.18) to model many periodic signals is studied.
Chapter 6

Sufficiency of Second-Order Nonlinear ODE for Modeling Periodic Signals

6.1 Introduction

IN the previous chapter the second-order nonlinear ODE model described by (5.18) was suggested to model periodic signals. A natural questions that follow directly: “When it is sufficient to use second-order ODE model?” or “What types of periodic signals can be modeled by (5.18)?”. These questions were addressed in [122, 123] and detailed answers were provided. In this chapter a brief review of the main results of [122, 123] is given.

It is stressed that the contents of this chapter is not a contribution of the author. It is included to provide the reader with a complete picture of the problem treated in Part II of the thesis.

The work done in [122, 123] proceeds as follows. First, some assumptions of general validity on the modeled signal and its derivatives are given. Then smoothness conditions are introduced to guarantee that the phasor constructed from the periodic signal $y(t)$ and its first derivative $\dot{y}(t)$, where $t$ denotes continuous time, i.e

$$y(t) = \begin{pmatrix} y(t) \\ \dot{y}(t) \end{pmatrix} \in \mathbb{R}^2, \quad (6.1)$$

represents a periodic orbit without any intersections. It is then proved that there exists an ODE that has the given signal as one solution. The provided conditions are emphasized by a phase plane geometrical intuition. Finally, the uniqueness of the solution of the ODE model is treated.

The chapter is organized as follows. General assumptions on the periodic signal are introduced in Section 6.2. Section 6.3 presents the smoothness conditions necessary to guarantee that the periodic orbit does not contain any intersections. In Section 6.4, necessary conditions on $y(t)$ to be a solution of the second-order ODE model are given. Uniqueness of the solution of the second-order ODE model is addressed in Section 6.5. Section 6.6 discusses the issue of insufficiency of the second-order ODE model. Finally, conclusions are given in Section 6.7.
6.2 General Assumptions on the Modeled signal

The work of [122, 123] begins by assuming some conditions on the modeled signal that can be satisfied for many signals. It is assumed that the periodic signal $y(t)$ fulfills Assumption 5.1 and the following assumptions:

Assumption 6.1. The signal $y(t)$ is not corrupted by any disturbance.

Assumption 6.2. The signal $y(t)$ is twice continuously differentiable.

It follows from Assumption 6.2 that $\dot{y}(t + T) = \dot{y}(t), \forall t \in \mathbb{R}$, where $T$ denotes the period. Hence, the signal $y(t)$ can be represented as the first coordinate of the closed curve generated by $\mathbf{y}(t)$.

6.3 Smoothness of the Modeled Orbit

As mentioned in the introduction of this chapter, the objective is to construct an autonomous second-order ODE system such that $y(t)$ is a solution of this ODE system. Hence, the selected state vector of the ODE model must contains all necessary informations to construct the closed orbit representing the signal $y(t)$. This means that the generated closed orbit by $\mathbf{y}(t)$ should not contain any intersections such as the intersection at point $P_I$ in Fig. 6.1. In such cases, the evolution of the orbit after the point $P_I$ will depend on the evolution before $P_I$. Then, the further evolution of $y(t)$ can not be determined uniquely from one single point in $\mathbb{R}^2$.

A condition that excludes situations where periodic orbit $y(t)$ intersects itself is given in [122, 123] as follows:

Assumption 6.3. For all $y(t) \in S \subset \mathbb{R}^2$:

- $ \begin{pmatrix} y(t_1) \\ \dot{y}(t_1) \end{pmatrix} = \begin{pmatrix} y(t_2) \\ \dot{y}(t_2) \end{pmatrix} \Rightarrow t_1 = t_2 + kT, \quad k \in \mathbb{Z} $
where $Z$ denotes the set of all integers.

- $\exists \delta, L_1, L_2, \delta > 0, 0 < L_1 \leq L_2 < \infty$ such that

\[ |t_1 - t_2| < \delta \Rightarrow L_1 |t_1 - t_2| \leq \left\| \left( \begin{array}{c} y(t_1) \\ \dot{y}(t_1) \end{array} \right) - \left( \begin{array}{c} y(t_2) \\ \dot{y}(t_2) \end{array} \right) \right\| \leq L_2 |t_1 - t_2|. \]

The smoothness Assumption 6.3 ensures the exclusion of signals that construct intersected phase planes and any other degenerate cases such as cusps, corners and stops. This was given in implicit way in Assumption 5.3. Also, the following lemma was proved in [122, 123].

**Lemma 6.1.** Under Assumptions 6.2-6.3, the curve $y(t) \in \mathbb{R}^2$ is smooth and without self-intersections. Further, the speed $\sqrt{\dot{y}(t)\dot{y}(t)}$ of the curve $y(t)$ is strictly positive for all $t$ and satisfy the following relation

\[ 0 < L_1 \leq \sqrt{\dot{y}(t)\dot{y}(t)} \leq L_2 < \infty. \]

**Proof.** See [122, 123].

Since the speed of the curve $y(t)$ is proved to be positive and finite in Lemma 6.1, it follows that the velocity vector $\dot{y}(t)$ which is tangent to $y(t)$ varies smoothly and no jump phenomena can take place.

### 6.4 Conditions on $y(t)$ to Be a Solution

The next step is to find the conditions necessary for $y(t)$ to be a solution for the second-order ODE model. Selecting the state vector to be

\[
\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} y(t) \\ \dot{y}(t) \end{pmatrix},
\]

(6.4)

and differentiating (6.4) with respect to time $t$, cf. Assumption 6.2, gives

\[
\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ \dot{y}(t) \end{pmatrix}.
\]

(6.5)

This equation represents the second-order nonlinear ODE system suggested to model the periodic signal. This system is autonomous only if the second right hand side function $\dot{y}(t)$ can be expressed in terms of the states $x_1$ and $x_2$. This can be done by solving (6.4) w.r.t. the time $t$. The implicit function theorem, see [62] page 651, is useful in this case. The theorem is stated as follows.

**Theorem 6.1.** Assume that $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is continuously differentiable at each point $(x, z)$ of an open set $P \subset \mathbb{R}^n \times \mathbb{R}^m$. Let $(x_0, z_0)$ be a point in $P$ for which $f(x_0, z_0) = 0$ and for which the Jacobian matrix $[\partial f/\partial x](x_0, z_0)$ is nonsingular. Then there exist neighborhoods $U \subset \mathbb{R}^n$ of $x_0$ and $V \subset \mathbb{R}^m$ of $z_0$ such that for each $z \in V$ the equation $f(x, z) = 0$ has a unique solution $x \in U$. Moreover, this solution can be given as $x = g(z)$, where $g$ is continuously differentiable at $z = z_0$. 
Figure 6.2: Solving for the time in the phase plane plot. The point marked by ‘o’ corresponds to the point where solution is performed while points marked by ‘x’ and ‘+’ corresponds to alternative points with same $x_{2,0}$ and $x_{1,0}$, respectively.

Proof. See [2, 63].

Now consider a closed orbit like the one shown in Fig. 6.2 and a point $x_0 = (x_{1,0}, x_{2,0})^T$ on this orbit (marked by ‘o’). The intention now is to solve at $x_0$ to find $t$ as a function in $x_{1,0}$ and/or $x_{2,0}$. The state equation (6.4) can be written in the form

$$
\begin{pmatrix}
  f_1(x_1, t) \\
  f_2(x_2, t)
\end{pmatrix} = \begin{pmatrix} x_1 - y(t) \\
  x_2 - \dot{y}(t)
\end{pmatrix} = \begin{pmatrix} 0 \\
  0
\end{pmatrix}.
$$

(6.6)

Solving the first equation of (6.6) in the point $x_0$ gives the times $t_{1,i}$, $i = 1, \cdots, I$ (marked by ‘+’ in Fig. 6.2). Similarly, solving the second equation of (6.6) in the same point gives the times $t_{2,j}$, $j = 1, \cdots, J$ (marked by ‘x’ in Fig. 6.2). Furthermore, differentiating (6.6) with respect to time $t$, it follows that (cf. Assumption 6.2 and Lemma 6.1)

$$
\frac{\partial f_1(x_{1,0}, t_{1,i})}{\partial t} = -\dot{y}(t_{1,i}) \neq 0,
$$

(6.7)

$$
\frac{\partial f_2(x_{2,0}, t_{2,j})}{\partial t} = -\ddot{y}(t_{2,j}) \neq 0.
$$

(6.8)

Therefore, it can be concluded that for all $t_{1,i}$ and $t_{2,j}$ that fulfill (6.7) and (6.8), respectively, there are continuously differentiable functions $h_{1,i}$, $i = 1, \cdots, I$ and $h_{2,j}$, $j = 1, \cdots, J$ such that

$$
t = h_{1,i}(x_1), \ t \in U_{1,i}, \ x_1 \in V_{1,i}, \ i = 1, \cdots, I
$$

(6.9)

$$
t = h_{2,j}(x_2), \ t \in U_{2,j}, \ x_2 \in V_{2,j}, \ j = 1, \cdots, J
$$

(6.10)

where $U_{1,i}, V_{1,i}, U_{2,j}, V_{2,j}$ are the corresponding neighborhoods of $t_{1,i}$, $x_{1,0}$, $t_{2,j}$, $x_{2,0}$, respectively. Since any intersection possibility in the periodic orbit
was excluded by Assumption 6.3, there is only one \( i = i_1 \) and one \( j = j_1 \) such that \( t_{1,i_1} = t_{2,j_1} \) within the same period.

**Remark 6.1.** Note that by Lemma 6.1 \( \sqrt{\dot{y}^2(t) + \dot{y}^2(t)} \geq L_1 > 0 \). Hence Eq. (6.7) and/or Eq. (6.8) always holds. In this case, the valid equation(s) can be used to solve for \( t \).

The analysis done in [122, 123] proceeds by selecting a number of points \( \{(x_{1,k}, x_{2,k})^T\}_{k=1}^K \in S \), ordered clockwise or counterclockwise around the curve \( y(t) \), such that the resulting overlapped neighborhoods covers one complete period. Then the times \( \{t_k\}_{k=1}^K \) corresponding to these points are computed. This procedure was formulated in [122, 123] by the following assumptions:

**Assumption 6.4.** \( I_k \leq \bar{I} < \infty \), \( J_k \leq \bar{J} < \infty \), \( \forall (x_{1,k}, x_{2,k})^T \in S \), \( k = 1, \cdots, K \).

**Assumption 6.5.** \( U_{\rho(k),k} \cap U_{\rho(k+1),k+1} \neq \emptyset \), \( k = 1, \cdots, K - 1 \) and \( U_{\rho(K),K} \cap (U_{\rho(1),1} + T) \neq \emptyset \).

**Assumption 6.6.** \( \bigcup_{k=1}^K U_{\rho(k),k} \supset [\bar{t}, \bar{t} + T] \) for an appropriately selected \( \bar{t} \).

Here the quantities \( I_k \) and \( J_k \) corresponds to quantities \( I \) and \( J \) for \( x_0 \). Also, \( \{U_{\rho(k),k}\}_{k=1}^K \) and \( \{V_{\rho(k),k}\}_{k=1}^K \) are the neighbourhoods around the points \( \{(x_{1,k}, x_{2,k})^T\}_{k=1}^K \), where

\[
\rho(k) = \begin{cases} 
1 & \text{if Eq. (6.7) is used} \\
2 & \text{if Eq. (6.8) is used.}
\end{cases}
\]

Hence, it is concluded in [122, 123] from Assumptions 6.4-6.6 that there exists neighborhoods \( \bar{U}_{\rho(k),k} \subset U_{\rho(k),k} \) and \( \bar{V}_{\rho(k),k} \subset V_{\rho(k),k} \) such that

\[
\bar{U}_{\rho(k),k} \cap \bar{U}_{\rho(k+1),k+1} = \emptyset, \ k = 1, \cdots, K - 1 \tag{6.11}
\]

\[
\bar{U}_{\rho(K),K} \cap (\bar{U}_{\rho(1),1} + T) = \emptyset, \tag{6.12}
\]

\[
\bigcup_{k=1}^K \bar{U}_{\rho(k),k} = [\bar{t}, \bar{t} + T], \tag{6.13}
\]

\[
h_{\text{period}}(x_1, x_2) \triangleq t = \begin{cases} 
h_{1, i_1(k)}^k(x_1), & x_1 \in \bar{V}_{1,k}, \ \rho(k) = 1 \\
h_{2, j_1(k)}^k(x_2), & x_2 \in \bar{V}_{2,k}, \ \rho(k) = 2.
\end{cases} \tag{6.14}
\]

**Remark 6.2.** The object of defining the neighborhoods \( \bar{U}_{\rho(k),k} \) and \( \bar{V}_{\rho(k),k} \) is to remove the overlapping region between the underlying neighborhoods. This is because the solution around the underlying points must be the same in the overlapping region.

Now substituting by expression (6.14) in Eq. (6.5) leads in [122, 123] to the following theorem:
Theorem 6.2. Consider the periodic signal \( y(t) \). Under Assumption 5.1 and Assumptions 6.1-6.6, there exists a function \( h_{\text{period}}(x_1, x_2) \) and a second-order ordinary differential equation

\[
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{pmatrix} = \begin{pmatrix}
x_2 \\
\dot{y}(h_{\text{period}}(x_1, x_2))
\end{pmatrix},
\]

that has a solution given by \( (x_1, x_2)^T = (y(t), \dot{y}(t))^T \).

6.5 Uniqueness of the Solution

The solution provided by Theorem 6.2 may not be unique. Uniqueness of the solution follows in case \( h_{\text{period}}(x_1, x_2) \) is Lipschitz, cf. Section 1.5.3. The following two corollaries were concluded in [122, 123] from [62]:

**Corollary 6.3.** Assume that the conditions of Theorem 6.2 hold. Assume in addition that there are constants \( L_3 \) and \( L_4 \) such that

\[
\begin{align*}
\left| \dot{y}(h_{\text{period}}(x_1, x_2)) - \dot{y}(h_{\text{period}}(z_1, z_2)) \right| &\leq L_3 \left\| \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \right\|, \\
\left| \dot{y}(h_{\text{period}}(x_1^0, x_2^0)) \right| &\leq L_4,
\end{align*}
\]

\( \forall (x_1, x_2)^T, (z_1, z_2)^T \in R^2 \) and \( \forall t \in [t_0, t_1] \). Then there is a second-order ODE

\[
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{pmatrix} = \begin{pmatrix}
x_2 \\
\dot{y}(h_{\text{period}}(x_1, x_2))
\end{pmatrix}, \quad \begin{pmatrix} x_1(t_0) \\ x_2(t_0) \end{pmatrix} = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix} \in S,
\]

that has the unique solution \( (x_1, x_2)^T = (y(t), \dot{y}(t))^T, \ t \in [t_0, t_1] \).

**Corollary 6.4.** Assume that the conditions of Theorem 6.2 hold. Assume in addition that there is a constant \( L_5 \) such that

\[
\left| \dot{y}(h_{\text{period}}(x_1, x_2)) - \dot{y}(h_{\text{period}}(z_1, z_2)) \right| \leq L_5 \left\| \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \right\|,
\]

\( \forall (x_1, x_2)^T, (z_1, z_2)^T \in D \subset R^2 \) and \( \forall t \geq t_0 \). Let \( W \) be a compact subset of the domain \( D, (x_1^0, x_2^0)^T \in W \), and suppose that it is known that every solution of

\[
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{pmatrix} = \begin{pmatrix}
x_2 \\
\dot{y}(h_{\text{period}}(x_1, x_2))
\end{pmatrix}, \quad \begin{pmatrix} x_1(t_0) \\ x_2(t_0) \end{pmatrix} = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix} \in S,
\]

(which exists) lies entirely in \( W \). Then this second-order ODE has the unique solution \( (x_1, x_2)^T = (y(t), \dot{y}(t))^T, \ t \in [t_0, t_1] \).
6.6 When a Second-Order ODE Model Is Insufficient

From the previous discussion, it can be noticed that higher order ODE models are needed in case the periodic orbit intersect itself in $\mathbb{R}^2$. The work of [122, 123] suggested to increase the order of the ODE in steps (one order more in each step). In this case, the extra state is selected as the next higher derivative of the periodic signal. For example, in case of extension from second-order ODE, the state variable $x_3 = \ddot{y}(t)$ is added. If the phase plot in $\mathbb{R}^3$ still contains intersections, a fourth state $x_4 = d^3y(t)/dt^3$ is added and this procedure is repeated until all intersections disappear.

The analysis done in Section 6.3 and Section 6.4 was repeated for the general order case in [122, 123] and the following theorem was proved.

**Theorem 6.5.** Consider a periodic signal $y(t)$. Under Assumption 5.1, Assumption 6.2, Assumptions 6.5-6.6 and the following assumptions:

**Assumption 6.7.** The signal $y(t)$ is $n + 1$ times continuously differentiable.

**Assumption 6.8.** For all $\begin{pmatrix} y \\ \vdots \\ y^{(n)} \\ y^{(n)}(t_1) \end{pmatrix} \in S^{n+1} \subset \mathbb{R}^{n+1}$:

- $\begin{pmatrix} y(t_1) \\ \vdots \\ y^{(n)}(t_1) \end{pmatrix} = \begin{pmatrix} y(t_2) \\ \vdots \\ y^{(n)}(t_2) \end{pmatrix} \Rightarrow t_1 = t_2 + kT, \ k \in \mathbb{Z}$.
- $\exists \delta, L_1, L_2, \delta > 0, 0 < L_1 \leq L_2 < \infty$ such that

$$|t_1 - t_2| < \delta \Rightarrow L_1|t_1 - t_2| \leq \left\| \begin{pmatrix} y(t_1) \\ \vdots \\ y^{(n)}(t_1) \end{pmatrix} - \begin{pmatrix} y(t_2) \\ \vdots \\ y^{(n)}(t_2) \end{pmatrix} \right\| \leq L_2|t_1 - t_2|.$$

**Assumption 6.9.** $I_{m,k} \leq \bar{I} < \infty, \ \forall (x_{1,k} \cdots x_{n+1,k})^T \in S^{n+1}$, $m = 1, \cdots, n+1, \ k = 1, \cdots, K$.

there exists a function $h_{\text{period}}(x_1, \cdots, x_{n+1})$ and an $n + 1$th order ODE

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \\ \dot{x}_{n+1} \end{pmatrix} = \begin{pmatrix} x_2 \\ x_3 \\ \vdots \\ x_{n+1} \\ x^{(n)}(h_{\text{period}}(x_1, \cdots, x_{n+1})) \end{pmatrix},$$

that has a solution given by $(x_1 \cdots x_{n+1})^T = (y(t) \cdots y^{(n)}(t))^T$.

**Proof.** See [122, 123].
6.7 Conclusions

The main results of the work done in [122, 123] have been introduced in this chapter. The results show that the approach of modeling periodic signals using second-order nonlinear ODE model is sufficient to model many periodic signals provided that the phase plot generated from the modeled signal and its first derivative does not contain any intersections. In case that the mentioned phase plot contains any intersections, a higher order ODE model is necessary to accurately model the periodic signal. The new ODE model is constructed by adding extra states to the original state vector. The extra states are chosen to be equal to the higher order derivatives of the modeled signal.
Chapter 7

Least Squares and Markov Estimation of Periodic Signals

7.1 Introduction

The linear regression model is a very common concept in statistics. It is the simplest type of parametric models. The linear regression model can be written as

\[ Y(t) = \phi^T(t) \theta, \]  

(7.1)

where \( Y(t) \) is a measurable quantity, \( \phi(t) \) is an \( n \)-vector of known quantities and \( \theta \) is an \( n \)-vector of unknown parameters to be estimated. The elements of the vector \( \phi(t) \) are often called *regressors* while \( Y(t) \) is called the regressed variable.

Introducing the equation errors as

\[ \varepsilon(t) = Y(t) - \phi^T(t) \theta, \]  

(7.2)

the least squares (LS) estimate of the unknown vector \( \theta \) (often called the *parameter vector*) is defined as the vector \( \hat{\theta}_{LS}^N \) that minimize the loss function

\[ V(\theta) = \frac{1}{2} \sum_{t=1}^{N} \varepsilon^2(t), \]  

(7.3)

where \( N \) is the number of available measurements. The LS estimate \( \hat{\theta}_{LS}^N \) is given by (see [99])

\[ \hat{\theta}_{LS}^N = \left[ \sum_{t=1}^{N} \phi(t)\phi^T(t) \right]^{-1} \left[ \sum_{t=1}^{N} \phi(t)Y(t) \right]. \]  

(7.4)

This estimate is expected to be *consistent* (unbiased) provided that the equation errors, \( \varepsilon(t) \), are zero-mean white.

In this chapter, an LS estimation algorithm is developed for the problem of modeling periodic signals using second-order nonlinear ODE’s. As will be shown, the developed LS algorithm gives biased estimates, especially at low signal to noise ratios (SNRs). One of the reasons that contribute to this bias is the invalidity of the whiteness assumption on the equation errors \( \varepsilon(t) \). Therefore,
a Markov estimation algorithm is introduced to compensate for this bias by using an estimate for the covariance matrix of the residuals. The performance of the two algorithms are illustrated in a simulation study.

The chapter is organized as follows. Section 7.2 discusses the LS estimation algorithm. The Markov estimation algorithm is introduced in Section 7.3. Section 7.4 presents a simulation study for the LS and the Markov estimation algorithms. Conclusions appear in Section 7.5.

7.2 The Least Squares Algorithm

A number of different algorithms can now be derived from the second-order ODE model of Chapter 5. In order to formulate the model in a linear regression form, two approximations are needed, cf. Eq. (5.17).

1. As $x_1 = y(t)$ is not known, use the estimate
   \[ \tilde{x}_1(kh) = z(kh). \]  
   \hspace{1cm} (7.5)

2. As $x_2 = \dot{y}(t)$ is not known, use the estimate
   \[ \tilde{x}_2(kh) = z(kh) - \frac{z(kh - h) - z(kh)}{h}. \]  
   \hspace{1cm} (7.6)

**Remark 7.1.** More general differentiating filters than (7.6) can be used. Reasonable differentiating filters are of the form
   \[ \tilde{H}(q) = \frac{(1 - q^{-1})}{h} H(q), \]  
   \hspace{1cm} (7.7)

with $H$ a low pass filter ($q^{-1}z(kh) = z(kh - h)$) of unity static gain. In this case (7.6) is replaced by
   \[ x_d(kh + h) = K x_d(kh) + L z(kh), \]
   \[ \tilde{x}_2(kh) = M x_d(kh) + d z(kh). \]  
   \hspace{1cm} (7.8)

Note that the filter output in (7.8) is only an estimate of $x_2(kh)$ since it is constructed from a measured signal. See [25] for more details about differentiating filters. In particular, the choice $K = 0$, $L = 1/h$, $M = -1/h$, $d = 1/h$ gives the estimate (7.6).

**Remark 7.2.** In the following, for notational convenience the dependence on $h$ is omitted assuming $h$ equals one unit time. This means that an integer $k$ can be used as time variable.

To proceed, note that Eqs. (5.29)-(5.30) result in the model
   \[ x_1(k) - x_1(k - 1) = x_2(k), \]  
   \hspace{1cm} (7.9)
   \[ x_2(k) - x_2(k - 1) = \phi^T (x_1(k), x_2(k)) \theta, \]  
   \hspace{1cm} (7.10)
7.2. The Least Squares Algorithm

where
\[ \phi^T(x_1(k), x_2(k)) = (1 \cdots x_2^M(k) \cdots x_1^T(k) \cdots x_2^T(k)x_1^N(k)) \], \quad (7.11)\]
\[ \theta = (\theta_{0,0} \cdots \theta_{0,M} \cdots \theta_{L,0} \cdots \theta_{L,M})^T. \quad (7.12)\]

By replacing the states of Eqs. (7.9)-(7.10) with the estimates given in (7.5) and (7.6), the second state equation (7.10) results in
\[ \hat{x}_2(k) = \hat{x}_2(k) - \hat{x}_2(k-1) = \phi^T(\hat{x}_1(k), \hat{x}_2(k)) \theta + \varepsilon(k). \quad (7.13) \]

The first state equation (7.9) is not needed. The expression (7.13) follows by performing a Taylor series expansion of \( \phi^T(x_1(k), x_2(k)) \) around \((\hat{x}_1(k), \hat{x}_2(k))^T \). In (7.13) the combined regression error, \( \varepsilon(k) \), has been introduced. It can not be expected to be white since the Taylor series expansion in (7.13) produces a sum of noise samples that are delayed with a varying number of sampling periods. This means that the least squares estimator will, in the end, be biased. However, for relatively high signal to noise ratios the accuracy could still be good. Note that (7.13) is a linear regression with the parameter vector \( \theta \) as unknown.

Assuming that data are available at times \( k-N, \cdots, k \) and defining the vectors and matrices
\[ Z_N = (\hat{x}_2(k) \cdots \hat{x}_2(k-N+1))^T, \quad (7.14) \]
\[ \Phi_N = \begin{pmatrix} \phi^T(\hat{x}_1(k), \hat{x}_2(k)) \\ \vdots \\ \phi^T(\hat{x}_1(k-N+1), \hat{x}_2(k-N+1)) \end{pmatrix}, \quad (7.15) \]
\[ \varepsilon_N = (\varepsilon(k) \cdots \varepsilon(k-N+1))^T, \quad (7.16) \]
the following regression equation results
\[ Z_N = \Phi_N \theta + \varepsilon_N. \quad (7.17) \]

Hence the LS estimate \( \hat{\theta}_N^{LS} \) is given by (see [99])
\[ \hat{\theta}_N^{LS} = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T Z_N. \quad (7.18) \]

Note that it is not numerically sound to solve the least squares problem directly using the normal equations that appear implicitly in (7.18). Instead a formulation as an over-determined system of equations should be used, see [99] for more details.

Remark 7.3. The LS estimate in (7.18) is expected to give considerably accurate models at high signal to noise ratios (SNRs) and further research is needed to extend the operating region toward low SNRs. Realizing that Eq. (7.17) is
an error-in-variables (EIV) problem since both $Z_N$ and the columns of $\Phi_N$ have noise contributions, the possibility of using the total least squares (TLS), see [107, 108], has been studied. The idea was to calculate the standard deviation of the noise disturbance on each column of the data matrix $[Z_N \Phi_N]$ and to design a weighting matrix $G$ to have similar noise contributions on the columns of the data matrix. In this case, Eq. (7.17) becomes

$$Z_N \approx (\Phi_N G) \bar{\theta},$$

(7.19)

where $\bar{\theta}$ is a weighted parameter vector and the estimate of the true parameter vector follows as

$$\hat{\theta} = G \bar{\theta}^{TLS}.$$  

(7.20)

Using the weighted TLS did not lead to improvements compared to the LS estimate at moderate SNR. Hence, it is relevant to neglect noise effects and use the LS type of solution.

### 7.3 The Markov Estimate

The least squares estimate $\hat{\theta}^{LS}_N$ given in (7.18) is expected to be approximately unbiased in case the regression error vector $\varepsilon_N$ is white, i.e. its covariance matrix has the form $\sigma^2 I$. This is not the case here due to the Taylor series expansion of (7.13). If the covariance matrix

$$R_N = E[\varepsilon_N \varepsilon_N^T]$$

(7.21)

is known, the Markov estimate, also known as the BLUE (best linear unbiased estimate) will satisfy [99]

$$\hat{\theta}^{Markov}_N = \left(\Phi_N^T R_N^{-1} \Phi_N\right)^{-1} \Phi_N^T R_N^{-1} Z_N.$$  

(7.22)

This estimator rests on the assumption that the mean of the noise is zero and that the system is in the model set. The above is not exactly true here, but the estimator can of course be applied anyway. Note that stationarity is assumed in the computation of $R_N$ in order to avoid a dependence on $k$.

Using Eqs. (5.12), (7.5)-(7.6), the left hand side of Eq. (7.13) can be written as

$$\hat{x}_2(k) = \hat{x}_2(k) - \hat{x}_2(k-1)
= z(k) - 2z(k-1) + z(k-2)
= y(k) - 2y(k-1) + y(k-2) + e(k) - 2e(k-1) + e(k-2)
= \hat{y}(k) - \hat{y}(k-1) + \hat{e}(k) - \hat{e}(k-1).$$

(7.23)

Hence Eq. (7.13) becomes

$$\hat{y}(k) - \hat{y}(k-1) + \hat{e}(k) - \hat{e}(k-1) = \phi^T (\hat{x}_1(k), \hat{x}_2(k)) \theta + \varepsilon(k).$$

(7.24)

In the following, it will be shown how an approximate value of $R_N$ can be computed under the following assumption:
Assumption 7.1. The contributions to $\varepsilon(k)$ from the regressor vector $\phi^T(\hat{x}_1(k), \hat{x}_2(k))$ do not dominate over the noise that enter linearly in (7.24), i.e. the equation error $\varepsilon(k)$ fulfills

$$
\varepsilon(k) \approx \hat{\varepsilon}(k) - \hat{\varepsilon}(k-1),
$$

where $e(k)$ is the white noise in Eq. (5.12).

The idea is then to neglect the regressor vector contribution and only use the parts of the noise that enter linearly. Therefore based on Assumption 7.1, (7.16) becomes

$$
\varepsilon_N = \begin{pmatrix}
\hat{\varepsilon}(k) - \hat{\varepsilon}(k-1) \\
\vdots \\
\hat{\varepsilon}(k - N + 1) - \hat{\varepsilon}(k - N)
\end{pmatrix},
$$

(7.25)

Now, an extended state space model can be constructed for the regression error vector $\varepsilon_N$ using the differentiating filter (7.8) and time delays. Consider the following state vector

$$
X(k) = \begin{pmatrix}
X_1^T(k) \\
X_2^T(k)
\end{pmatrix}^T,
$$

(7.26)

where

$$
X_1(k) = x_d(k),
$$

(7.27)

$$
X_2(k) = \begin{pmatrix}
\hat{\varepsilon}(k - 1) \\
\vdots \\
\hat{\varepsilon}(k - N)
\end{pmatrix}^T.
$$

(7.28)

Then, under Assumption 7.1, the matrix $R_N$ can be computed from the following asymptotically stable stochastic discrete time system

$$
X(k+1) = AX(k) + Be(k),
$$

$$
\varepsilon_N(k) = DX(k) + Ee(k).
$$

(7.29)

Remark 7.4. The system (7.29) is intended to model the correlation properties of the noise that appears in the computation of the matrix $R_N$. For this reason, the superposition principle is used to subtract the signal from the input to the differentiator filter so that only the noise remains as an input. Note that the dynamics of the system (7.29) need to account for the dynamics of the differentiating filter, and the delays of the states needed in order to generate the vector $\varepsilon_N$ given by (7.25).

Let $n_d$ denote the order of the differentiating filter (7.8). Then, the block matrices of (7.29) are given by

$$
A = \begin{pmatrix}
K & 0_{n_d \times N} \\
D_1 & S_N
\end{pmatrix},
$$

(7.30)

$$
B = \begin{pmatrix}
L \\
0_{(N-1) \times 1}
\end{pmatrix},
$$

(7.31)
where $0_{i \times j}$ denotes an empty matrix of $i$ rows and $j$ columns. Also, the $N \times N$ dimensional shift-matrix
\[
S_N = \begin{pmatrix}
0 & \cdots & 0 \\
1 & 0 & \cdots \\
0 & 1 & 0 & \cdots \\
\vdots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & 1 & 0
\end{pmatrix},
\] (7.34)
generates $(N-1)$ delayed states. The $N \times (N + n_d)$ matrix $D$ is built up from the blocks
\[
D_1 = \begin{pmatrix} M \\ 0_{(N-1) \times n_d} \end{pmatrix},
\] (7.35)
\[
D_2 = \begin{pmatrix}
-1 & 0 & 0 & \cdots & 0 & 0 \\
1 & -1 & 0 & \cdots & 0 & 0 \\
0 & 1 & -1 & \cdots & 0 & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & 1 & 0 & -1
\end{pmatrix}.
\] (7.36)
It is now straightforward to compute $R_N$. First the stationary state covariance matrix of (7.29), i.e.
\[
P_N = \mathbb{E}[X(k)X^T(k)]
\] (7.37)
is computed as the unique positive definite symmetric solution of the stationary Lyapunov equation (see e.g. [55])
\[
P_N = A P_N A^T + \sigma^2 B B^T.
\] (7.38)
Then, from (7.29)
\[
R_N \approx D P_N D^T + \sigma^2 E E^T.
\] (7.39)

**Remark 7.5.** In case the residual covariance matrix $R_N$ is singular or ill-conditioned, the matrix $R_N + \Phi_N \Phi_N^T$ may be better conditioned to inversion than $R_N$. The following (theoretically equivalent) formula can be used instead of (7.22) (see [99]);
\[
\hat{\theta}_N^{\text{Markov}} = \left[ \Phi_N^T \left( R_N + \Phi_N \Phi_N^T \right)^{-1} \Phi_N \right]^{-1} \Phi_N^T \left( R_N + \Phi_N \Phi_N^T \right)^{-1} Z_N,
\] (7.40)
where $Y^\dagger$ denotes the pseudoinverse of the matrix $Y$.

### 7.4 Numerical Examples

In the following two sections a detailed simulation study for the LS algorithm and the Markov estimation algorithm is given.
7.4. Numerical Examples

7.4.1 The LS Algorithm Simulation Study

The Van der Pol oscillator [62] was selected as the underlying system in Examples 7.1-7.3 of this simulation study. The oscillator was described by

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{bmatrix} = \begin{bmatrix}
x_2 \\
-x_1 + 2(1 - x_1^2)x_2
\end{bmatrix}.
\]  

(7.41)

The Matlab routine `ode45` was used to solve (7.41). The initial state of (7.41) was selected as \( (x_1(0) \ x_2(0))^T = (0 \ 1)^T \). All results below are based on data runs of length \( N = 10^4 \). The measured signal was in all examples selected as the first state with white Gaussian noise added. The differentiated signal was obtained by applying a simple first order difference (Euler forward approximation). The signals generated from (7.41) with \( h = 0.01 \) s are shown in Fig. 7.1.

Example 7.1. The effect of the sampling period on the LS algorithm.

The noise level was selected in this example so that a SNR of 60 dB was obtained. The estimated model used third-degree polynomials, i.e. \( L = M = 3 \). The LS algorithm was run for different sampling periods. As a measure of performance,

\[
V = \frac{\|\hat{\theta}_N - \theta^o\|_2}{\|\theta^o\|_2}
\]  

(7.42)

was computed and plotted as a function of \( h \) in Fig. 7.2. In (7.42), \( \theta^o \) denotes the true parameter vector.

The minimum of the plot can be explained as follows. For small values of \( h \), the numerical differentiation results in a large noise amplification. The error caused by the regression vector then becomes dominant. For large values of \( h \) the most pronounced effect is instead modeling errors introduced by the Euler forward numerical discretization scheme. The former error decreases with \( h \) while the latter error in general increases with \( h \), and hence an optimal \( h \) may
exist. A more detailed study for the effect of the choice of sampling interval on the performance of the LS algorithm is given in Chapter 12.

**Example 7.2. The effect of the SNR on the LS algorithm.**

In this example the setup was the same as in Example 7.1, with the exceptions that the SNR was varied and that $L = M = 2$. The sampling periods were selected to be $h = 0.075$ s, $h = 0.1$ s and $h = 0.2$ s. The performance measure (7.42) was plotted for the three cases in Fig. 7.3. As a further illustration a phase plane plot and a periodogram appear in Fig. 7.4. Figure 7.4 was generated using $h = 0.075$ s.

The plots of the performance measure (7.42) in Fig. 7.3 are consistent with the explanation given in response to Fig. 7.2. It is clear from Fig. 7.3 that the LS algorithm provides good estimates only at high SNR. Also the minimum of the plots can be similarly explained as in Example 7.1 by the fact that the total bias is a contribution of discretization errors and random noise errors,
Example 7.3. The effect of over-parameterization on the LS algorithm.

Simulations were run for the polynomial degrees 2, 3, 4 and 5 using SNR=60 dB and \( h = 0.1 \) s. The performance as expressed by (7.42) is illustrated in Fig. 7.5. The result indicates that the LS algorithm can cope with degrees that are incorrect with at least one unit.

Example 7.4. The effect of under-modeling on the LS algorithm.

In order to study the effect of under-modeling the LS algorithm was used to identify the undamped physical pendulum system

\[
\ddot{y} = -\frac{g}{\lambda} \sin(y).
\]

Here \( g = 9.81 \text{ m/s}^2 \) and \( \lambda = 1 \text{ m} \). The sampling time was selected as \( h = 0.01 \) s. The system was initiated at rest with a 90 degree angle with the vertical. All simulations used SNR=50 dB. The system was identified with polynomial degrees ranging from 2 to 7 as illustrated in Fig. 7.6. The results of Fig. 7.6 show that the accuracy of the obtained model improves as the polynomial degree increases. This is consistent with the fact that the accuracy of the polynomial approximation of \( \sin(y) \) increases with the polynomial degree.

### 7.4.2 The Markov Estimate Compared to the LS Estimate

The Van der Pol oscillator given in Eq. (7.41) was selected also as the underlying system in Examples 7.5-7.6 of this simulation study. 1000 data samples were generated with \( h = 0.03 \) s. The differentiated signal was obtained in this simulation by applying the fourth order low-pass Butterworth filter [86]

\[
H(q^{-1}) = \frac{0.662 + 2.648q^{-1} + 3.972q^{-2} + 2.648q^{-3} + 0.662q^{-4}}{1 + 3.181q^{-1} + 3.861q^{-2} + 2.112q^{-3} + 0.4383q^{-4}}, \quad (7.44)
\]
Figure 7.6: Phase plane plot for different polynomial degrees. System (solid) and identified model (dashed).
with a bandwidth \((0.9\pi/h)\) rad/sec on the signal generated from a simple first order difference (Euler backward approximation). The Bode plot of the differentiation filter is shown in Fig. 7.7.

**Example 7.5.** This example illustrates the superior performance of the Markov algorithm as compared to the LS algorithm. In this example the estimated model used second-degree polynomials. Both algorithms were run for different SNRs. As a measure of performance, Eq. (7.42) was computed and plotted as a function of the SNR in Fig. 7.8. Also, the phase plots for the estimated models at a SNR of 60 dB and the true system are given in Fig. 7.9.

It can be noticed from Fig. 7.8 that even the Markov estimation algorithm gives significantly better estimates than the LS algorithm, the Markov estimates at low and moderate SNR are practically not useful since \(V \approx 1\). This is due to the fact that only 1000 data samples were used and further increase in the data length was not possible due to high computational complexity of the algorithm (cf. Chapter 13).
Example 7.6. This example illustrates the effect of over-parameterization on the performance of the Markov estimation algorithm as compared to the LS algorithm. Simulations were run for the polynomial degrees 2, 3, 4 and 5 using a SNR of 50 dB. The performance as expressed by Eq. (7.42) is illustrated in Fig. 7.10. The results indicate that the Markov estimator is less sensitive to over-parameterization than the LS estimator.

7.5 Conclusions

The main idea of this chapter is to model periodic signals as being generated by second-order nonlinear ODE’s. A linear in the parameters polynomial model was used in order to derive a least squares algorithm. This algorithm was tested in a simulation study and was shown to work well for high signal to noise ratios. Robustness with respect to over- and under-modeling was also demonstrated. The sampling period was found to be an important tuning parameter.

It is concluded from the simulation study of the LS algorithm that further research is needed to extend the operating region towards lower signal to noise ratios and to get ride of the bias. For this reason, a Markov estimation algorithm was developed in this chapter. An extended state space representation was used in order to derive an estimate of the noise correlation properties of the required prefiltering. The suggested algorithm results in improved parameter estimates as compared to the LS estimation algorithm. However, the Markov estimation algorithm is computationally demanding. From a practical point of view, both the LS and the Markov algorithms are more suitable for high SNR cases. This is due to the fact that the accuracy of the finite difference approximations used is degraded significantly at low SNR.
Chapter 8

Periodic Signal Modeling with Kalman Filters

8.1 Introduction

In Chapter 7 the least squares and the Markov estimation algorithms were introduced for the approach of modeling periodic signals using second-order nonlinear ODE’s. These algorithms are off-line or batch identification algorithms, i.e. data were assumed to be collected and available before identification process starts. In the identification process, the batch data available are used to find a second-order ODE model that resembles the periodic signal generation. Needless to say that the LS algorithm can easily be written as a recursive algorithm, see [99] for details.

In 1960, R. E. Kalman published his celebrated paper [61] describing a recursive solution to the discrete-data linear filtering problem. Since that time, the Kalman filter has been used in many applications such as aerospace, target tracking, marine and satellite navigation, see e.g. [24, 39, 127].

Kalman’s solution, known as the Kalman filter (KF), is a set of mathematical equations that provide an efficient computational (recursive) means to estimate the state of a process that is governed by a linear stochastic difference equation, in a way that minimizes the mean of the squared error. The filter is very powerful since it supports estimations of the system (past, present, and future) states even when the precise nature of the modeled system is unknown.

In practice, it is very often the case that the process to be estimated and/or the measurement relationship to the process are non-linear. Then it is appropriate to use the extended Kalman filter (EKF), that linearizes around the previous state, instead of the KF. The EKF approach is to apply the standard KF (for linear systems) to nonlinear systems with additive white noise by continually updating this linearization, starting with an initial guess. In other words, a linear Taylor approximation of the system function is only considered at the previous state estimate and that of the observation function at the corresponding predicted position. This approach gives a simple and efficient algorithm to handle nonlinear models. However, convergence to the true states may not be obtained if the initial guess is poor or if the disturbances are so large that the linearization is inadequate to describe the system.

In this chapter, recursive identification (on-line) algorithms, such as the Kalman filter and the extended Kalman filter are developed. In this case the
second-order ODE model of the periodic signal is updated at each time instance some new data sample becomes available. In practice, recursive algorithms may also be used for a batch of data as done in this chapter.

The chapter is organized as follows. The KF algorithm is presented in Section 8.2. Section 8.3 introduces the EKF algorithm. In Section 8.4, a simulation study is presented. Finally, conclusions follow in Section 8.5.

### 8.2 The Kalman Filter (KF)

Similarly to Chapters 5-7, the following ODE model is assumed to generate the periodic signal.

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{bmatrix} = \begin{bmatrix}
x_2(t) \\
\phi^T(x_1(t), x_2(t)) \theta
\end{bmatrix},
\]

\[y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\
x_2(t) \end{bmatrix}. \tag{8.1}\]

Here, the vectors \(\phi^T(x_1(t), x_2(t))\) and \(\theta\) are given by

\[
\phi^T(x_1(t), x_2(t)) = \begin{bmatrix} 1 & \cdots & x_2^M(t) & \cdots & x_1^L(t) & \cdots & x_2^L(t) & \cdots & x_1^M(t) \end{bmatrix},
\]

\[\theta = \begin{bmatrix} \theta_{0,0} & \cdots & \theta_{0,M} & \cdots & \theta_{L,0} & \cdots & \theta_{L,M} \end{bmatrix}^T. \tag{8.2}\]

The model (8.1) is discretized to allow for the implementation of recursive algorithms. In the recursive case the Euler forward numerical integration scheme, cf. Section 5.6, is used and the discretized model becomes

\[
x_1(kh + h) = x_1(kh) + hx_2(kh),
\]

\[
x_2(kh + h) = x_2(kh) + h\phi^T(x_1(kh), x_2(kh))\theta. \tag{8.4}\]

The Kalman filter is obtained by introduction of the parameters as states, using a random walk assumption, see [96]. Other alternatives than a random walk assumption are possible. However, the choice used here is standard in the literature. Note that the construction of the regression vector \(\phi^T(x_1(kh), x_2(kh))\) of (8.2) is based on the exact states, which are not available in practice. These quantities must therefore be approximated in all practical algorithms. This procedure differs between the KF and the EKF algorithms, treated in this chapter.

The idea of the Kalman filter algorithm is to replace the first state appearing in the regression vector by the measured output signal, \(z(kh)\), a choice that follows from the definition of the output vector in Eq. (8.1). Due to the selected structure, the second state is replaced by an estimate of the derivative of the measured output signal. This latter signal, denoted \(\hat{\dot{z}}(kh)\), is obtained from a differentiating filter, cf. Eqs. (7.5)-(7.6) and Remark 7.1.

The state vector of the Kalman filter is now augmented with the estimated parameter vector as follows

\[
x(kh) = \begin{bmatrix} x_1(kh) & x_2(kh) & \theta^T \end{bmatrix}^T. \tag{8.5}\]
Hence, using equations (8.1), (8.4) and (8.5) result in an extended state space model given by
\[
x(kh + h) = F(h, z(kh), \hat{z}(kh))x(kh) + w(kh),
\]
\[
z(kh) = Hx(kh) + e(kh),
\] (8.6)

where
\[
F(h, z(kh), \hat{z}(kh)) = \begin{pmatrix} 1 & h & 0 \\ 0 & 1 & h \phi^T(z(kh), \hat{z}(kh)) \\ 0 & 0 & I_{(M+1)(L+1)} \end{pmatrix},
\] (8.7)
\[
H = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}.
\] (8.8)

In order to complete the Kalman filter description, noise properties and initial values need to be set. The measurement noise properties follow from Assumption 5.2. Because of the nonlinear data dependence in \( \phi^T(z(kh), \hat{z}(kh)) \), the system noise is not Gaussian. However, the Kalman filter can be applied anyway. The performance of the algorithm can be expected to be close to that of an exact Kalman filter, provided that the following conditions hold with a sufficient accuracy.

**Assumption 8.1.** The process noise vector satisfy the following:
\[
w(kh) = \begin{pmatrix} w_1(kh) & w_2(kh) & w_θ^T(kh) \end{pmatrix}^T \in \mathcal{N}(0, R_1),
\]
\[
E[w(kh)w^T(kh + jh)] = \delta_{j,0}R_1.
\]

**Assumption 8.2.** The initial values
\[
\hat{x}(t_0|t_0 - h) = E[x(t_0)|x(t_0 - h)],
\]
\[
P(t_0|t_0 - h) = E[(x(t_0) - \hat{x}(t_0|t_0 - h))(x(t_0) - \hat{x}(t_0|t_0 - h))^T],
\]
define a Gaussian distribution of the prior state.

Now, the Kalman filter is given by, see [96];
\[
K(kh) = P(kh|kh - h)H^T\left(HP(kh|kh - h)H^T + r_2\right)^{-1}
\]
\[
\hat{x}(kh|kh) = \hat{x}(kh|kh - h) + K(kh)(z(kh) - H\hat{x}(kh|kh - h))
\]
\[
P(kh|kh) = P(kh|kh - h) - P(kh|kh - h)H^T \times \left(HP(kh|kh - h)H^T + r_2\right)^{-1}HP(kh|kh - h)
\] (8.9)
\[
\hat{x}(kh + h|kh) = F(h, z(kh), \hat{z}(kh))\hat{x}(kh|kh)
\]
\[
P(kh + h|kh) = F(h, z(kh), \hat{z}(kh))P(kh|kh)F^T(h, z(kh), \hat{z}(kh)) + R_1.
\]
8.3 The Extended Kalman Filter (EKF)

The extended Kalman filter differs from the Kalman filter in one crucial point - the state propagation matrix is built up from the recursively estimated states rather than directly from measured data, i.e. Eqs. (8.6)-(8.7) are replaced by

\begin{align*}
\mathbf{x}(kh+h) &= \mathbf{F}(h, \hat{\mathbf{x}}_1(kh), \hat{\mathbf{x}}_2(kh)) \mathbf{x}(kh) + \mathbf{w}(kh), \\
z(kh) &= \mathbf{H} \mathbf{x}(kh) + \mathbf{e}(kh), \\
\mathbf{F}(h, \hat{\mathbf{x}}_1(kh), \hat{\mathbf{x}}_2(kh)) &= \begin{pmatrix}
1 & h \\
0 & 1 & \mathbf{h}\phi^T(\hat{\mathbf{x}}_1(kh), \hat{\mathbf{x}}_2(kh)) \\
0 & 0 & I_{(M+1)(L+1)}
\end{pmatrix} .
\end{align*}

(8.10)

Following [96], the extended Kalman filter recursions are now given by

\begin{align*}
\mathbf{K}(kh) &= \mathbf{P}(kh|kh-h) \mathbf{H}^T (\mathbf{H} \mathbf{P}(kh|kh-h) \mathbf{H}^T + r_2)^{-1} \\
\hat{\mathbf{x}}(kh|kh) &= \hat{\mathbf{x}}(kh|kh-h) + \mathbf{K}(kh)(z(kh) - \mathbf{H} \hat{\mathbf{x}}(kh|kh-h)) \\
\mathbf{P}(kh|kh) &= \mathbf{P}(kh|kh-h) - \mathbf{P}(kh|kh-h) \mathbf{H}^T \\
&\times \left( \mathbf{HP}(kh|kh-h) \mathbf{H}^T + r_2 \right)^{-1} \mathbf{HP}(kh|kh-h) \\
\mathbf{P}(kh+h|kh) &= \mathbf{F}(h, \hat{\mathbf{x}}(kh|kh)) \hat{\mathbf{x}}(kh|kh) \\
\hat{\mathbf{F}}(kh) &= \frac{\partial \mathbf{F}(h, \mathbf{x})}{\partial \mathbf{x}} \bigg|_{\mathbf{x}=\hat{\mathbf{x}}(kh|kh)} \\
\mathbf{P}(kh+h|kh) &= \mathbf{F}(kh) \mathbf{P}(kh|kh) \hat{\mathbf{F}}^T(kh) + \mathbf{R}_1.
\end{align*}

(8.12)

Remark 8.1. The modifications as compared to the Kalman filter may seem minor. However, it is stressed that they are not. First, since there are no nonlinear transformations of noisy measurements, Assumption 8.1 can now be expected to hold exactly, and no significant bias problems are expected. Secondly, the dynamics of Eqs. (8.10)-(8.11) is highly nonlinear. It is in fact polynomial. Hence instability phenomena and even finite escape time effects may come into play. See [62] for further details.

8.4 Simulation Study

The proposed algorithms were tested with data generated by a Van der Pol oscillator as described in [62]. This system can here be described by the ODE

\begin{align*}
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{pmatrix} &= \begin{pmatrix}
x_2 \\
-x_1 + 2(1-x_1^2)x_2
\end{pmatrix} .
\end{align*}

(8.13)

Data sets of length $3 \times 10^4$ samples were generated by solving (8.13) with the Matlab routine \texttt{ode45}, using $h = 0.01$ s. Initial states were chosen as $(x_1(0), x_2(0))^T = (0, 1)^T$. In the KF estimation algorithm, $\hat{\mathbf{x}}(kh)$ was obtained using the Euler center approximation, see Section 5.6.
Example 8.1. Bias properties of the KF and the EKF algorithms.

In this example the bias properties of the KF and the EKF are investigated. White Gaussian noise was added to obtain data with a signal to noise ratio (SNR) of 15 dB. The algorithms were initialized with \( \hat{x}_1(0) \hat{x}_2(0) \hat{\theta}^T(0) = (-0.5 \ 0.5 \ 0)^T \) and the remaining parameters were selected as \( P(0 - h) = 10I \), \( R_1 = 10^{-5}I \) and \( r_2 = 1 \). The parameter vector was selected according to Eq. (8.3) with \( L = M = 2 \). The simulation results are available in Figures 8.1-8.3, where the Kalman filter and the extended Kalman filter are compared.

\[ \begin{align*}
\hat{x}_1(t) & \quad \hat{x}_2(t) \\
\hat{\theta}(t) & \quad \hat{\theta}(t)
\end{align*} \]

**Figure 8.1:** Parameter convergence.

\[ \begin{align*}
\hat{x}_1(t) & \quad \hat{x}_2(t) \\
\hat{\theta}(t) & \quad \hat{\theta}(t)
\end{align*} \]

**Figure 8.2:** True (solid) and estimated (dashed) phase plane plots. The KF (left) and the EKF (right) [SNR=15dB].
It can be concluded from Figures 8.1-8.3 that the EKF algorithm has a superior performance. It appears to be unbiased, which is not true for the KF algorithm. However, note that when a polynomial degree of three was tried at this low SNR, the EKF became unstable, while the KF still delivered usable results. This is in line with the predictions of Remark 8.1.

**Example 8.2. The effect of the SNR on the KF and the EKF algorithms.**

In this example the performance of the KF and the EKF is studied as the SNR changes. The goal is to see the effect of the SNR on the KF bias encountered in Example 8.1. As a measure of performance,

\[ V = \frac{\| \Theta_N - \Theta^o \|_2}{\| \Theta^o \|_2} \]  

was computed and plotted as a function of the SNR in Fig. 8.4.

The results of Fig. 8.4 indicate that the bias of the KF estimates decreases as the SNR increases. It can be concluded that the performance of the KF algorithm is quite close to the performance of the EKF algorithm for moderate and high SNRs. Also, Fig. 8.4 shows that the EKF is more suitable for low SNR cases than the LS and the Markov algorithms, see Figures 7.3 and 7.8.

The phase plots for the true and estimated phase plots at SNR = 40 dB are given in Fig. 8.5.
**8.4. Simulation Study**

**Figure 8.4:** The effect of the SNR on the performance of the KF and the EKF algorithms.

**Figure 8.5:** True (solid) and estimated (dashed) phase plane plots. The KF (left) and the EKF (right) [SNR=40dB].
Example 8.3. The effect of over-parameterization on the KF and the EKF algorithms.

To study the effect of the polynomial degree on the performance of the KF and the EKF algorithms, simulations were run for the polynomial degrees 2, 3 and 4 using SNR=50 dB. The performance as expressed by (8.14) is illustrated in Fig. 8.6.

The results indicate that the KF algorithm appears to be slightly more robust against over-parameterization than the EKF algorithm. Also, when a polynomial degree of 5 was tried, the EKF algorithm became unstable.

8.5 Conclusions

In this chapter, two recursive algorithms, namely the Kalman filter algorithm and the extended Kalman filter algorithm, were developed for modeling periodic signals using second-order nonlinear ODE’s. The KF, although biased at low signal to noise ratios, performed robustly and can be recommended, e.g., for initial value generation for the EKF. The latter algorithm showed a performance that was superior to the KF especially for low SNRs. On the other hand, the EKF algorithm is more sensitive to over-parameterization than the KF algorithm and instability problems are encountered at high polynomial degrees.

Also, the EKF is found to be more suitable for low SNRs cases than the LS and the Markov algorithms of Chapter 7. This is due to the fact that in the EKF algorithm the states are estimated recursively and simultaneously with the parameter vector instead of using the noisy measured signal and its derivatives.
Chapter 9

Maximum Likelihood Estimation of Periodic Signals

9.1 Introduction

Let \( X \) denote the vector of observations of a stochastic variable and let \( p(X, \theta) \) denote the probability density function (pdf) of \( X \). The form of \( p(X, \theta) \) is assumed to be known. The parameter vector \( \theta \), which completely describes the pdf, is to be estimated. The maximum likelihood (ML) estimate of \( \theta \) is defined by

\[
\hat{\theta}_{ML} = \arg \max_{\theta} p(X, \theta).
\] (9.1)

Thus with the ML approach the value of \( \theta \) is chosen which makes the data most plausible as measured by the likelihood function \( p(X, \theta) \).

In practice, it is often impossible to find an analytical solution for (9.1). This is due to the fact that the model is complex and involves many parameters and/or complex probability functions. Hence, Eq. (9.1) is usually solved numerically using an optimization (or minimization) algorithm. Another common practical problem when implementing the ML estimation algorithm is the existence of local minima. Hence, it is very important to have a good initial condition for the minimization algorithm to avoid convergence to a local minimum instead of the true (global) minimum.

In this chapter a ML algorithm is developed for periodic signal estimation using second-order nonlinear ODE’s. The approach is analyzed by derivation and solution of a system of ODE’s that describes the evolution of the Cramér-Rao bound (CRB) over time. This allows the theoretically achievable accuracy of the proposed method to be assessed in the ideal case where the signals can be exactly described by the imposed model.

The chapter is organized as follows. Section 9.2 discusses the details on the ODE models including a definition of the parameterization. Section 9.3 introduces the ML algorithm, while Section 9.4 presents the CRB. The chapter ends with a simulation study and conclusions in Sections 9.5 and 9.6, respectively.

9.2 The ODE models

In order to model the periodic signal \( y(t) \) that is defined in Section 5.3, the following two models are considered.
• Model I:

\[
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{pmatrix} = \begin{pmatrix}
f_1(x_1(t), x_2(t), \theta_1) \\
f_2(x_1(t), x_2(t), \theta_2)
\end{pmatrix},
\]

\[z(t) = \begin{pmatrix} c_1 & c_2 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + e(t), \tag{9.2}\]

where \( (x_1(t) \ x_2(t))^T \) is the state vector and \( (c_1 \ c_2) \) is the vector containing the selected output weighting factors that generate \( y(t) \) when multiplied with the state vector. Furthermore

\[
\theta = \begin{pmatrix} \theta_1^T \\ \theta_2^T \end{pmatrix}^T, \tag{9.3}
\]

is the unknown parameter vector.

• Model II:

\[
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{pmatrix} = \begin{pmatrix} x_2(t) \\
f_2(x_1(t), x_2(t), \theta_2)
\end{pmatrix},
\]

\[z(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + e(t). \tag{9.4}\]

As mentioned in Chapter 5, Model II is considered more restrictive than Model I since Model II depends only on the parameters of the second right hand side function \( f_2(x_1(t), x_2(t), \theta_2) \). The analysis of Chapter 6 also suggests that one right hand function is sufficient. However, this choice of model structure is by no means a necessity. Future developments may very well result in further motivation for the use of Model I. In order to obtain results of a general validity, the treatment of this chapter will therefore deal with the two models.

Expanding the right hand side of the state equations of (9.2) as well as the second state equation of (9.4) in terms of a polynomial model and discretizing the two models using the Euler forward approximation (cf. Section 5.6), Model I and Model II become (for \( t = k h \))

• Model I:

\[
x_1(t + h) = x_1(t) + \phi_1^T(x_1(t), x_2(t), h)\theta_1,
\]

\[
x_2(t + h) = x_2(t) + \phi_2^T(x_1(t), x_2(t), h)\theta_2,
\]

\[z(t) = c_1 x_1(t) + c_2 x_2(t) + e(t). \tag{9.5}\]

• Model II:

\[
x_1(t + h) = x_1(t) + h x_2(t),
\]

\[
x_2(t + h) = x_2(t) + \phi_2^T(x_1(t), x_2(t), h)\theta_2,
\]

\[z(t) = x_1(t) + e(t). \tag{9.6}\]

Here

\[
\phi_1(x_1(t), x_2(t), h) = h \begin{pmatrix} 1 & \cdots & x_2^{M_1}(t) & \cdots & x_1^{L_1}(t) & \cdots & x_1^{L_1}(k)x_2^{M_1}(t) \end{pmatrix}^T,
\]

\[
\phi_2(x_1(t), x_2(t), h) = h \begin{pmatrix} 1 & \cdots & x_2^{M_2}(t) & \cdots & x_1^{L_2}(t) & \cdots & x_1^{L_2}(k)x_2^{M_2}(t) \end{pmatrix}^T. \tag{9.7}\]
\[
\theta_1 = ( \theta_{1,0,0} \ldots \theta_{1,0,M_1} \ldots \theta_{1,L,0} \ldots \theta_{1,L,M_1} )^T, \\
\theta_2 = ( \theta_{2,0,0} \ldots \theta_{2,0,M_2} \ldots \theta_{2,L,0} \ldots \theta_{2,L,M_2} )^T.
\]

(9.8)

**Remark 9.1.** Note that the difference between Model I and Model II is not very important as such. The complexity and obtainable accuracy depends more on the number of parameters used for modeling, than on the models themselves. The two models thus offer two possibilities for signal modeling and the required number of parameters for each structure can be expected to depend on the application. Also, note that \( L_2 \) and \( M_2 \) are not necessarily the same for Model I and Model II.

### 9.3 The Maximum Likelihood Method

Before proceeding with the development of the criterion function, it is noted that equations (9.5) and (9.6) can be simultaneously treated by a use of the model

\[
x(t + h) = Fx(t) + \Phi(x(t), h)\theta, \quad x(t_0) = x_{t_0}, \\
z(t) = Hx(t) + e(t),
\]

(9.9)

where \( x = (x_1(t) \quad x_2(t))^T \). This follows by the selections

\[
F = \begin{pmatrix}
1 & 0 \\
0 & 1 
\end{pmatrix}, \\
\Phi(x(t), h) = \begin{pmatrix}
\phi_1^T(x(t), h) \\
0 \\
\phi_2^T(x(t), h)
\end{pmatrix},
\]

\[
\theta = (\theta_1^T \quad \theta_2^T)^T, \\
H = (c_1 \quad c_2),
\]

and

\[
F = \begin{pmatrix}
1 & h \\
0 & 1
\end{pmatrix}, \\
\Phi(x(t), h) = \begin{pmatrix}
0 \\
\phi_2^T(x(t), h)
\end{pmatrix},
\]

(9.11)

where (9.10) and (9.11) correspond to Model I and Model II, respectively.

Proceeding with the statement of the maximum likelihood criterion it is observed that the measurement disturbance in Eq. (9.9) is assumed to be Gaussian, i.e. Assumption 5.2 is assumed to hold. Note also that the initial values \( x_{t_0} = x(t_0) \) of the states need to be non-stochastic and estimated together with the unknown non-stochastic parameter vector for the problem to be meaningful.
with Gaussian distributions. It follows that the likelihood function equals
\[
p(Z^N, \theta, x_{t_0}) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(z(t_0) - Hx_{t_0})^2}{2\sigma^2}\right) \times \\
\frac{1}{(2\pi)^{N/2}\sigma^N} \prod_{k=1}^{N} \exp\left[-\frac{(z(t_0 + kh) - Hx(t_0 + kh, \theta, x_{t_0}))^2}{2\sigma^2}\right],
\]
(9.12)

where \(Z^N\) denotes the set of all measurements and \(N\) denotes the number of measurements. The dependence on the parameter vector and the initial value enter implicitly in Eq. (9.12) via the modeled states \(x(t_0 + kh, \theta, x_{t_0})\) at the sampling instances. The generation of the model states is perhaps best understood by referring to the criterion minimization strategy that is outlined in the end of this section. There the model (9.9) is iterated to generate a complete state trajectory (of model states). This is done for the fixed set of initial values \(x_{t_0}\) and parameters \(\theta\), that correspond to one specific iteration of the criterion minimization procedure.

Taking logarithms, scaling and changing signs, it is straightforward to see that the maximization of (9.12) is equivalent to the following minimization problem;
\[
\left( \begin{array}{c} \hat{\theta}_{ML}^T \\ \hat{x}_{t_0}^T \end{array} \right)^T = \arg \min_{\theta, x_{t_0}} \hat{L}(\theta, x_{t_0}),
\]
(9.13)

where
\[
\hat{L}(\theta, x_{t_0}) = (z(t_0) - Hx_{t_0})^2 + \sum_{k=1}^{N} (z(t_0 + kh) - Hx(t_0 + kh, \theta, x_{t_0}))^2.
\]
(9.14)

Typically, the minimization of the criterion \(\hat{L}(\theta, x_{t_0})\) has to be carried out in several steps since (9.13) seldom has a unique minimum point and since a dense grid search in a high dimensional parameter space is generally not feasible. Note that if a grid search would be applied, the model (9.9) would have to be iterated from \(t_0\) to \(t_0 + Nh\) once for each grid point, using the parameters and initial values defined by the grid point.

One tentative multistep criterion minimization procedure could follow the steps:

1. Use any of the low complexity algorithms such as the LS method of Chapter 7 or the KF of Chapter 8, possibly followed by the EKF, to compute initial estimates of \(\theta\) and \(x_{t_0}\).

2. Possibly perform a grid search around this initial estimate in order to refine it further.

3. Perform a final gradient or Gauss-Newton iterative search, using the refined initial estimate as initial values to the iteration.
Obviously, there are many ways to vary this theme. It is, e.g., possible to compute gradients by forming differences numerically between two trajectories of (9.9). Alternatively, sensitivity derivatives could be computed analytically and integrated.

9.4 The Cramér-Rao Bound

The accuracy of the maximum likelihood estimates obtained will depend on the evolution of the trajectories of the nonlinear ODE that describes the system. The observability of the parameters will vary with the state of the ODE during the orbit. The approach taken to compute the CRB in this chapter allows for handling of orbits that are only asymptotically periodic, a situation that naturally arises because of transient phenomena. These effects are modeled by the inclusion of the initial values of the ODE in the CRB computation.

Remark 9.2. It should be noted that the CRB is limited to a study of accuracy in the ideal case where the system (or signal) can be exactly described by the imposed model, by at least one value of the parameter vector. The CRB hence covers one aspect of the analysis of the performance of the proposed method, by assessment of the theoretically achievable accuracy. One inevitable consequence of the selected model structure is that the accuracy study by means of the CRB is not being made against the conventional parameters like frequency, phase and amplitude, rather the accuracy of the parameters of the polynomial expansion of the model is studied.

Remark 9.3. The “unbiased” CRB is used in the computations below. The estimates for the initial values may however be biased - no proof of the opposite has been provided. However, the part of the CRB that describes the parameter vector should not be affected by this. The reason is that the transient effects of the initial values fade away and hence the asymptotic estimates can not be affected. This means that the unbiased CRB is useful for studying the accuracy of the asymptotic estimates, possibly excluding the estimates of the initial values. See [99, 110] for more details on the biased CRB.

The model used in the CRB computation is selected as the most general of the models used for estimation, i.e. Model I described by (9.2) in combination with (9.3). The CRB for Model II described by (9.4) follows by exclusion of the relevant matrix blocks of the obtained result for Model I. The model used for the development of the CRB is hence

$$\begin{align*}
\begin{pmatrix}
\dot{x}_1(t, \theta, \psi) \\
\dot{x}_2(t, \theta, \psi)
\end{pmatrix}
= & \begin{pmatrix}
f_1(x_1(t, \theta, \psi), x_2(t, \theta, \psi), \theta_1) \\
f_2(x_1(t, \theta, \psi), x_2(t, \theta, \psi), \theta_2)
\end{pmatrix}, \\
z(t, \theta, \psi) = & \begin{pmatrix} c_1 & c_2 \end{pmatrix} \begin{pmatrix} x_1(t, \theta, \psi) \\
x_2(t, \theta, \psi) \end{pmatrix} + \epsilon(t),
\end{align*}$$

with the initial values

$$\begin{pmatrix} x_1(t_0) \\
x_2(t_0) \end{pmatrix} = \begin{pmatrix} \psi_1 \\
\psi_2 \end{pmatrix} = \psi.$$

9.4. The Cramér-Rao Bound
The measurements are assumed to be discrete time and available at times $t_1 + h, \ldots, t_1 + Nh$. The objective is now to compute (see e.g. [99])

$$\text{CRB}^{-1} = E \left[ L^T(t, \theta, \psi)L(t, \theta, \psi) \right], \quad (9.17)$$

where

$$L(t, \theta, \psi) = \left( \log L_{\theta_1}, \log L_{\theta_2}, \log L_{\psi_1}, \log L_{\psi_2} \right). \quad (9.18)$$

As usual, the CRB is evaluated for the true parameter vector. The subscripts are used to denote partial differentiation. Since the measurement noise is Gaussian by Assumption 5.2, the likelihood function, $L(\theta, \psi)$, is up to a constant given by

$$-\log L(\theta, \psi) = \frac{1}{2\sigma^2} \sum_{k=1}^{N} \left( z(t_1 + kh) - c_1 x_1(t_1 + kh, \theta, \psi) - c_2 x_2(t_1 + kh, \theta, \psi) \right)^2. \quad (9.19)$$

The expectation of (9.17) can then be evaluated, by exploiting that for the true parameter vector, the output error is white Gaussian noise. The components of the expectation of (9.17) depend on the partial derivatives of the states with respect to the unknowns (the sensitivity derivatives). This follows directly from (9.17). The computation of the sensitivity derivatives can be performed by partial differentiation of both sides of the original differential equation (9.15) with respect to the unknowns. The result is a new set of ODE’s that need to be integrated together with the model (9.15), in order to compute the quantities needed in the computation of the components of (9.17). The initial values needed for the solution follow by partial differentiation of Eq. (9.16). The calculations are straightforward but tedious. The results are displayed in detail in Appendix 9.A.

The calculation of the CRB can now be summarized as follows:

1. Solve the system of differential equations given by (9.15) together with (9.33)-(9.40), using the initial conditions (9.16) and (9.41). This system is complete and can be solved with any high accuracy numerical routine for solving ODE’s. The solution is computed on $[t_0, t_{\text{end}}] \supseteq [t_1 + h, t_1 + Nh]$. 

2. Compute the elements of the Fisher information matrix (9.23)-(9.32) from the computed trajectories.

3. Invert (9.17) to finalize the computation.

### 9.5 Simulation Study

The focus of this simulation study is on accuracy. Thus, other important issues like convergence, local minima and the use of multiple algorithms for initialization purposes (outlined in Section 9.3) are not covered. The two main purposes are to study the theoretical accuracies achievable by means of the
CRB and to compare the maximum likelihood method performance to this bound. The following model of the Van der Pol oscillator was considered in this study
\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -x_1 + 2(1 - x_1^2)x_2 \quad (9.20)
\end{align*}
\]

The model structure given by (9.4) was used, and the parameters \(\theta_0, 1, \theta_1, 0\) and \(\theta_2, 1\) were estimated, together with the initial values. The remaining parameters were fixed to 0. This means that 3 parameters plus 2 initial values are estimated by the maximum likelihood algorithm and are also included in the CRB calculation. The reason for restricting the number of parameters was a need to restrict the run time of the simulations since convergence of the steepest descent algorithm (9.21) required more than 2000 iterations (this is not an uncommon number in practice, see [70]). The components corresponding to \(\theta_1\) in (9.17) disappear since here the CRB was computed from the restricted model, i.e., Model II. The state initial values were \(x_1(0) = x_2(0) = 0\).

The maximum likelihood criterion (9.13) was minimized with the following steepest descent algorithm
\[
\begin{align*}
\begin{pmatrix} \theta_{n+1} \\ x_{t_0,n+1} \end{pmatrix} &= \begin{pmatrix} \theta_{n} \\ x_{t_0,n} \end{pmatrix} \\
&- \frac{\left(\triangle \hat{L}(\theta_n, x_{t_0,n})\right)^T \left(\triangle \hat{L}(\theta_n, x_{t_0,n})\right)}{\left(\triangle \hat{L}(\theta_n, x_{t_0,n})\right)^T \mathcal{H}(\theta_n, x_{t_0,n}) \left(\triangle \hat{L}(\theta_n, x_{t_0,n})\right)} \triangle \hat{L}(\theta_n, x_{t_0,n}).
\end{align*}
\]
(9.21)

Here \(\mathcal{H}(\theta_n, x_{t_0,n})\) denotes the Hessian of \(\hat{L}(\theta_n, x_{t_0,n})\) and \(\triangle \hat{L}(\theta_n, x_{t_0,n})\) is the gradient of \(\hat{L}(\theta_n, x_{t_0,n})\). The reader is referred to [70], page 150 and page 154 for further details. As stated above, the focus is on accuracy. Hence, to avoid problems with local minima of the criterion function, the algorithm was initiated close to the true parameter vector \(\left(\begin{array}{c} \theta^0 \\ x_{t_0}^0 \end{array}\right)^T\).

**Example 9.1. The signal to noise ratio study.**

This example evaluates the CRB and the performance of the maximum likelihood method as a function of the SNR of the data. To evaluate the performance, 50 Monte-Carlo experiments, each using a data length of 2000 samples, were conducted for a number of different SNRs. The algorithm (9.21) was initiated according to
\[
\begin{pmatrix} \theta \\ x_{t_0} \end{pmatrix} = \begin{pmatrix} \theta^0 \\ x_{t_0}^0 \end{pmatrix} - 5\sigma(SNR, N),
\]
(9.22)

where \(\sigma(SNR, N)\) is the standard deviation as predicted by the CRB evaluated at specific SNR and data length \(N\). Note again that each iteration of (9.21) requires a solution of the ODE and the corresponding sensitivity derivatives that appear in Appendix 9.A.
Figure 9.1: Performance of the ML method vs. SNR. CRB (solid), MSE (dashed) and the Monte-Carlo 95% confidence interval (dotted).
The results of the evaluation appear in Fig. 9.1. The CRB is plotted solid while the performance of the maximum likelihood method is plotted dashed. The 95% confidence interval, see [99] page 579, of 50 Monte-Carlo experiments is plotted dotted. It can be seen that the (relevant) parameters are estimated with accuracies that are close to the CRB. This is to be expected, considering the fact that the maximum likelihood method is known to reach the CRB asymptotically under mild conditions. It can also be observed that the closeness to the CRB begins to deteriorate below SNRs of 10 dB. Finally, the CRBs roll off linearly with the SNR in a log-log plot. This is consistent with the fact that the standard deviation enters as a factor in the CRB.

Example 9.2. The data length study.

This example evaluates the CRB and the performance of the maximum likelihood method as a function of the number of samples available for estimation. As compared to Example 9.1, the following differences apply. First, all results are evaluated for a fixed SNR of 30 dB. Secondly, the data lengths were chosen so as to include an integer number of complete periods. The reason for this is to provide estimates that are balanced with respect to the state space signal energy available for estimation. Unless this would be the case, the maximum likelihood method can be expected to be biased for finite data sets, thus distorting the intended accuracy assessment.

The results of the evaluation appear in Fig. 9.2. It can be observed that the accuracy of the estimated parameters of the ODE model approaches the CRB when the available amount of data increases. This is consistent with the conjecture that the maximum likelihood method in this case should approach the CRB. The performance for the initial values is worse, cf. Remark 9.3. It can even be observed that the accuracy of the algorithm for the initial values gets worse when the data set is increased above a certain number of samples. This is believed to be a result of the fact that the effect of the initial values on the signals decay with time - hence the signal energy available for estimation of the initial values are located at the beginning of the data set. Therefore the overall signal to noise ratio for the estimation of these parameters starts to decrease after a certain period of time since only noise enters the estimation algorithm as far as the initial values are concerned. The result could then be a drift of these parameters. Note also that the initial values are not expected to be unbiased since the signal energy related to the initial values is finite, even when the data length turns to infinity. Hence, efficiency of the maximum likelihood method for these parameters cannot be expected.

Some further comments related to the simulations are in order. First, it is very important to apply the ODE solvers in a correct manner. One problem encountered was the fact that the inaccuracy of the Matlab ODE solvers increased when the high order ODE’s related to the model and the sensitivity derivatives were solved. Numerical instability was another problem. The problems were solved by explicit control of the required inaccuracy, and by the use of different ODE solvers for the estimated model and the gradient calculations. See Section 13.3.7 for details.
Figure 9.2: Performance of the ML method vs. data length. CRB (solid), MSE (dashed) and the Monte-Carlo 95% confidence interval (dotted).
9.6 Conclusions

Based on the second-order nonlinear ODE model for modeling periodic signals, a maximum likelihood (ML) method and the Cramér-Rao bound (CRB) for the selected model structure have been derived in this chapter. The CRB is computed via the solution of a large system of ordinary differential equations. The ML algorithm was tested in a simulation study and a comparison to the CRB was performed. The ML algorithm seems to give a performance very close to the CRB, approaching the CRB when the number of samples increases.

9.6. Conclusions

The elements of the CRB are first evaluated, exploiting the symmetry of (9.17). The elements become, using the whiteness of the noise

\[
E \left[ \left( \log \mathcal{L}_{\theta_1} \right)^T \left( \log \mathcal{L}_{\theta_1} \right) \right] = \\
\frac{1}{\sigma^2} \sum_{k=1}^{N} \left( \begin{array}{c} x_1(t_k + kh, \theta, \psi) \\ x_2(t_k + kh, \theta, \psi) \end{array} \right)_{\theta_1}^T \left( \begin{array}{cc} c_1 & c_1 \\ c_2 & c_2 \end{array} \right)^T \left( \begin{array}{c} x_1(t_k + kh, \theta, \psi) \\ x_2(t_k + kh, \theta, \psi) \end{array} \right)_{\theta_1},
\]

(9.23)

\[
E \left[ \left( \log \mathcal{L}_{\theta_2} \right)^T \left( \log \mathcal{L}_{\theta_2} \right) \right] = \\
\frac{1}{\sigma^2} \sum_{k=1}^{N} \left( \begin{array}{c} x_1(t_k + kh, \theta, \psi) \\ x_2(t_k + kh, \theta, \psi) \end{array} \right)_{\theta_2}^T \left( \begin{array}{cc} c_1 & c_1 \\ c_2 & c_2 \end{array} \right)^T \left( \begin{array}{c} x_1(t_k + kh, \theta, \psi) \\ x_2(t_k + kh, \theta, \psi) \end{array} \right)_{\theta_2},
\]

(9.24)

\[
E \left[ \left( \log \mathcal{L}_{\psi_1} \right)^T \left( \log \mathcal{L}_{\psi_1} \right) \right] = \\
\frac{1}{\sigma^2} \sum_{k=1}^{N} \left( \begin{array}{c} x_1(t_k + kh, \theta, \psi) \\ x_2(t_k + kh, \theta, \psi) \end{array} \right)_{\psi_1}^T \left( \begin{array}{cc} c_1 & c_1 \\ c_2 & c_2 \end{array} \right)^T \left( \begin{array}{c} x_1(t_k + kh, \theta, \psi) \\ x_2(t_k + kh, \theta, \psi) \end{array} \right)_{\psi_1},
\]

(9.25)

\[
E \left[ \left( \log \mathcal{L}_{\psi_2} \right)^T \left( \log \mathcal{L}_{\psi_2} \right) \right] = \\
\frac{1}{\sigma^2} \sum_{k=1}^{N} \left( \begin{array}{c} x_1(t_k + kh, \theta, \psi) \\ x_2(t_k + kh, \theta, \psi) \end{array} \right)_{\psi_2}^T \left( \begin{array}{cc} c_1 & c_1 \\ c_2 & c_2 \end{array} \right)^T \left( \begin{array}{c} x_1(t_k + kh, \theta, \psi) \\ x_2(t_k + kh, \theta, \psi) \end{array} \right)_{\psi_2},
\]

(9.26)

\[
E \left[ \left( \log \mathcal{L}_{\theta_1} \right)^T \left( \log \mathcal{L}_{\theta_2} \right) \right] = \\
\frac{1}{\sigma^2} \sum_{k=1}^{N} \left( \begin{array}{c} x_1(t_k + kh, \theta, \psi) \\ x_2(t_k + kh, \theta, \psi) \end{array} \right)_{\theta_2}^T \left( \begin{array}{cc} c_1 & c_1 \\ c_2 & c_2 \end{array} \right)^T \left( \begin{array}{c} x_1(t_k + kh, \theta, \psi) \\ x_2(t_k + kh, \theta, \psi) \end{array} \right)_{\theta_2},
\]

(9.27)
E \left[ \frac{1}{\sigma^2} \sum_{k=1}^{N} \left( \begin{array}{c} (x_1(t_1 + kh, \theta, \psi))_{\phi_1} \\ (x_2(t_1 + kh, \theta, \psi))_{\phi_1} \end{array} \right) ^T \begin{array}{cc} c_1 \\ c_2 \end{array} \right] ^T \begin{array}{cc} c_1 \\ c_2 \end{array} \right] ^T \begin{array}{c} (x_1(t_1 + kh, \theta, \psi))_{\theta_2} \\ (x_2(t_1 + kh, \theta, \psi))_{\theta_2} \end{array} \right] = \left( \begin{array}{c} \frac{1}{\sigma^2} \sum_{k=1}^{N} \left( \begin{array}{c} (x_1(t_1 + kh, \theta, \psi))_{\phi_2} \\ (x_2(t_1 + kh, \theta, \psi))_{\phi_2} \end{array} \right) ^T \begin{array}{cc} c_1 \\ c_2 \end{array} \right) \right) ^T \begin{array}{c} (x_1(t_1 + kh, \theta, \psi))_{\theta_2} \\ (x_2(t_1 + kh, \theta, \psi))_{\theta_2} \end{array} \right] = \left( \begin{array}{c} \frac{1}{\sigma^2} \sum_{k=1}^{N} \left( \begin{array}{c} (x_1(t_1 + kh, \theta, \psi))_{\phi_1} \\ (x_2(t_1 + kh, \theta, \psi))_{\phi_1} \end{array} \right) ^T \begin{array}{cc} c_1 \\ c_2 \end{array} \right) \right) ^T \begin{array}{c} (x_1(t_1 + kh, \theta, \psi))_{\phi_2} \\ (x_2(t_1 + kh, \theta, \psi))_{\phi_2} \end{array} \right] = \left( \begin{array}{c} \frac{1}{\sigma^2} \sum_{k=1}^{N} \left( \begin{array}{c} (x_1(t_1 + kh, \theta, \psi))_{\phi_2} \\ (x_2(t_1 + kh, \theta, \psi))_{\phi_2} \end{array} \right) ^T \begin{array}{cc} c_1 \\ c_2 \end{array} \right) \right) ^T \begin{array}{c} (x_1(t_1 + kh, \theta, \psi))_{\phi_2} \\ (x_2(t_1 + kh, \theta, \psi))_{\phi_2} \end{array} \right] = \left( \begin{array}{c} \frac{1}{\sigma^2} \sum_{k=1}^{N} \left( \begin{array}{c} (x_1(t_1 + kh, \theta, \psi))_{\phi_2} \\ (x_2(t_1 + kh, \theta, \psi))_{\phi_2} \end{array} \right) ^T \begin{array}{cc} c_1 \\ c_2 \end{array} \right) \right) ^T \begin{array}{c} (x_1(t_1 + kh, \theta, \psi))_{\phi_2} \\ (x_2(t_1 + kh, \theta, \psi))_{\phi_2} \end{array} \right] 

This completes the derivation of the different elements of the Fisher information matrix. It remains to compute the partial derivatives of the states with respect to the unknowns, i.e. the sensitivity derivatives. Towards this end, it is noted that a straightforward partial differentiation of (9.15) results in:

\[
\frac{d}{dt} \left( f_1 (x_1, x_2, \theta_1) \right) \phi_1 = \left( f_1 (x_1, x_2, \theta_1) \right) x_1 \left( f_1 (x_1, x_2, \theta_1) \right) \phi_1 + \left( f_1 (x_1, x_2, \theta_1) \right) x_2 \left( f_1 (x_1, x_2, \theta_1) \right) \phi_1.
\]

\[
\frac{d}{dt} \left( f_1 (x_1, x_2, \theta_1) \right) \phi_2 = \left( f_1 (x_1, x_2, \theta_1) \right) x_1 \left( f_1 (x_1, x_2, \theta_1) \right) \phi_2 + \left( f_1 (x_1, x_2, \theta_1) \right) x_2 \left( f_1 (x_1, x_2, \theta_1) \right) \phi_2.
\]
9.A. Derivation of the Details of the CRB

The initial conditions to these equations follow by differentiation of (9.16).

\[
\begin{align*}
\frac{d}{dt} (x_2(t, \theta, \psi))_{\theta_1} &= (f_2(x_1, x_2, \theta_2))_{x_2} (x_1(t, \theta, \psi))_{\theta_1} \\
&\quad + (f_2(x_1, x_2, \theta_2))_{x_2} (x_2(t, \theta, \psi))_{\theta_1}, \\
(9.35) \\
\frac{d}{dt} (x_2(t, \theta, \psi))_{\theta_2} &= (f_2(x_1, x_2, \theta_2))_{x_1} (x_1(t, \theta, \psi))_{\theta_2} \\
&\quad + (f_2(x_1, x_2, \theta_2))_{x_2} (x_2(t, \theta, \psi))_{\theta_2} + (f_2(x_1, x_2, \theta_2))_{x_2}, \\
(9.36) \\
\frac{d}{dt} (x_1(t, \theta, \psi))_{\psi_1} &= (f_1(x_1, x_2, \theta_1))_{x_1} (x_1(t, \theta, \psi))_{\psi_1} \\
&\quad + (f_1(x_1, x_2, \theta_1))_{x_2} (x_2(t, \theta, \psi))_{\psi_1}, \\
(9.37) \\
\frac{d}{dt} (x_1(t, \theta, \psi))_{\psi_2} &= (f_1(x_1, x_2, \theta_1))_{x_1} (x_1(t, \theta, \psi))_{\psi_2} \\
&\quad + (f_1(x_1, x_2, \theta_1))_{x_2} (x_2(t, \theta, \psi))_{\psi_2}, \\
(9.38) \\
\frac{d}{dt} (x_2(t, \theta, \psi))_{\psi_1} &= (f_2(x_1, x_2, \theta_2))_{x_1} (x_1(t, \theta, \psi))_{\psi_1} \\
&\quad + (f_2(x_1, x_2, \theta_2))_{x_2} (x_2(t, \theta, \psi))_{\psi_1}, \\
(9.39) \\
\frac{d}{dt} (x_2(t, \theta, \psi))_{\psi_2} &= (f_2(x_1, x_2, \theta_2))_{x_1} (x_1(t, \theta, \psi))_{\psi_2} \\
&\quad + (f_2(x_1, x_2, \theta_2))_{x_2} (x_2(t, \theta, \psi))_{\psi_2}, \\
(9.40)
\end{align*}
\]

The initial conditions to these equations follow by differentiation of (9.16).

\[
\begin{align*}
(x_1(t_0))_{\theta_1} &= (x_2(t_0))_{\theta_1} = 0, \\
(x_1(t_0))_{\theta_2} &= (x_2(t_0))_{\theta_2} = 0, \\
(x_1(t_0))_{\psi_1} &= (x_2(t_0))_{\psi_2} = 1, \\
(x_1(t_0))_{\psi_2} &= (x_2(t_0))_{\psi_1} = 0. \\
(9.41)
\end{align*}
\]

This completes the derivation of the CRB.
Chapter 10

Periodic Signal Modeling Based on Fully Automated Spectral Analysis

10.1 Introduction

It was shown in Chapters 5-9 that periodic signals can be modeled by means of second-order nonlinear ordinary differential equations (ODE’s). The right hand side function of the ODE model is parameterized in terms of known basis functions. The least squares (LS) algorithm developed in Chapter 7 for estimating the coefficients of these basis functions gives biased estimates, especially at low signal to noise ratios (SNRs).

The bias encountered in the least squares estimate is due to two reasons. First, the assumption on the residuals in the linear regression equation (7.13) to be white is not valid. In Chapter 7 this problem was addressed by estimating the noise covariance matrix and implementing a Markov estimate based algorithm. Second, derivatives of the modeled signal evaluated using finite difference approximations are highly contaminated with noise. Hence, the regressors and the regressed variable of the linear regression equation are also contaminated with noise, and the problem becomes an error-in-variables (EIV) problem, see [107, 108]. However, for good estimates of the derivatives of the modeled signal and/or relatively high SNRs the accuracy is good.

In this chapter a fully automated spectral analysis (ASA) technique, see [85, 94], is used to eliminate the noise contributions on the modeled signal and its derivatives. The ASA technique estimates the period length and the fundamental Fourier coefficients of the modeled signal. This allows an evaluation of accurate estimates for the derivatives of the modeled signal. This is expected to avoid noise amplification in the differentiation phase extending the operating region of the LS estimation algorithm toward low SNRs.

The chapter is organized as follows. Section 10.2 discusses the least squares estimate. The fully automated spectral analysis (ASA) technique is presented in Section 10.3. Section 10.4 gives a comparative simulation study between the least squares algorithm using the ASA technique (LS-ASA) and the least squares (LS) algorithm using finite difference approximation. Conclusions appear in Section 10.5.
10.2 The Least Squares Estimate

In Chapter 7 the second state equation of the ODE model (5.18) was formulated in the following linear regression form

$$\dot{x}_2(t) = \phi^T(x_1(t), x_2(t)) \theta, \quad (10.1)$$

where

$$\phi^T(x_1(t), x_2(t)) = \begin{pmatrix} 1 & \cdots & x_2^M(t) & \cdots & x_1^L(t) & \cdots & x_1^M(t) \end{pmatrix}, \quad (10.2)$$

$$\theta = \begin{pmatrix} \theta_{0,0} & \cdots & \theta_{0,M} & \cdots & \theta_{L,0} & \cdots & \theta_{L,M} \end{pmatrix}^T. \quad (10.3)$$

To estimate the parameter vector $\theta$ from (10.1), some approximations for $x_1(t)$, $x_2(t)$ and $\dot{x}_2(t)$ were needed. Replacing these states by their estimates, Eq. (10.1) resulted in (at $t = kh$)

$$\hat{\dot{x}}_2(kh) = \phi^T(\hat{x}_1(kh), \hat{x}_2(kh)) \theta + \varepsilon(kh). \quad (10.4)$$

The expression (10.4) followed by performing a Taylor series expansion of the regression vector $\phi^T(x_1(kh), x_2(kh))$ around $(\hat{x}_1(kh), \hat{x}_2(kh))^T$. Since the Taylor series expansion in (10.4) produces sums of noise samples that are delayed with a varying number of sampling periods, the combined regression error, $\varepsilon(kh)$, is not white.

The LS estimate (7.18) has been studied in Chapter 7 using

$$\hat{x}_1(kh) = z(kh) = y(kh) + \varepsilon(kh) \quad (10.5)$$

in addition to $\hat{x}_2(kh)$ and $\hat{\dot{x}}_2(kh)$ evaluated using finite difference approximation. It is shown in Chapter 7 that the LS algorithm gives accurate models at high SNRs and further research is needed to extend the operating region toward low SNRs. This is due to amplifying the measurement noise during differentiating the measured signal. In this chapter the ASA technique [94] is used to find more accurate estimates for the noise free periodic signal $x_1(t)$ and its derivatives, $x_2(t)$ and $\dot{x}_2(t)$. An accurate estimate $\tilde{x}_1(t)$ is found by eliminating noise corruption on the modeled signal. Also, more accurate estimates $\tilde{x}_2(t)$ and $\tilde{\dot{x}}_2(t)$ are found by differentiating $\tilde{x}_1(t)$ in the frequency domain. The ASA technique is discussed in the next section.

10.3 The Fully Automated Spectral Analysis Technique

The aim of the automated spectral analysis (ASA) algorithm is to estimate the spectrum of a periodic signal without requiring that:

- an integer number of periods is measured;
- the signal period equals an integer number of samples.

On the other hand the following properties are desired:
10.3. The Fully Automated Spectral Analysis Technique

- eliminating user choices: in other words, no user interaction is needed so that a fully automated procedure is available;
- the calculation time grows slowly with the record length (in the order of \( O(N \log N) \)) and this holds independent on the number of estimated harmonics.

In order to reach the previous goals the data record should contain at least 2 full periods + 48 samples (The 48 samples are needed to account for the shift in the FIR filter used in the algorithm). The spectrum of the signal is calculated up to \( 0.4 f_s \) (\( f_s = 1/h \)) with a relative error smaller than \( 10^{-5} \) of the peak value of the spectrum. The algorithm is explained in all details in [94]. Here only a brief introduction is given.

Remark 10.1. The reason for choosing the order of the FIR to be 48 is that it is the lowest order that achieves a systematic errors of 100 dB below the signal level in a frequency band up to \( 0.4 f_s \), see [94].

Consider the periodic signal \( z(t) \) given in (10.5) with basic frequency \( f_0 = 1/T \):

\[
  z(t) = \sum_{r=-F}^{F} Z_r e^{i2\pi f_0 t} 
\]

sampled at the time instances \( t = kh \), with \( f_0 F \leq 0.4 f_s \). \( Z_r = \bar{Z}_{-r} \) is the Fourier coefficient of the \( r \)th component (where \( \bar{ } \) denotes the complex conjugate). Note that \( T/h \) is not required to be a rational number. \( N \) equidistant measurements of \( z(kh) \) are made over more than 2 periods once the transients disappeared (the steady state solution is reached). Under these conditions the ASA technique allows to obtain estimates \( \hat{Z}_r \) of the Fourier coefficients \( Z_r \), together with an estimate of the variance

\[
  \sigma^2_Z(r) = E \left[ (\hat{Z}_r - E[\hat{Z}_r])(\hat{Z}_r - E[\hat{Z}_r]) \right].
\]

Remark 10.2. Note that \( F \) is chosen such that \( f_0 F \leq 0.4 f_s \) is fulfilled. This means that the bandwidth of the sampled data is restricted to 80% of the Nyquist frequency to allow upsampling and interpolation in the ASA technique without aliasing, see [85].

The procedure consists of two parts. In a first step, an initial estimate \( \hat{T}_0 \) of the period length \( T \) is made using correlation methods. In a second step this initial estimate is improved by minimizing a cost function \( V(T) \) (cf. Eq. (10.10)), and eventually the corresponding Fourier coefficients are calculated using the fast Fourier transform (FFT).

10.3.1 Initial Estimate \( \hat{T}_0 \)

The initial estimate of the period length is based on the autocorrelation \( R_{zz}(\tau) \) of \( z(t) \). The basic idea is to detect the distance between successive peaks in
\( R_{zz}(\tau) \). If a wide band signal with a flat amplitude spectrum is analyzed, this simple method gives a good estimate. However it fails in practice for a number of special cases. Since the method should be robust, it is refined in [94] to deal also with special signals like:

- \( z(t) \) is a beat signal (a narrow band signal) resulting in an strongly oscillating behaviour of \( R_{zz}(\tau) \) (instead of estimating the correct period, the period of this oscillation would be detected).
- \( z(t) \) is a beat signal with a strongly odd behaviour. These signals have a parasitic peak at half the period length.

### 10.3.2 Improved Estimate of the Period Length

The improved period will be obtained by minimizing a cost function \( V(T) \), that is defined below step by step.

1- Assume that the period of the signal is \( T \) (to be estimated). From the initial estimate it is known that the measurements cover more than \( M \in \mathbb{N} \) periods of the signal.

2- In the next step, the samples are interpolated with an equidistant grid with \( L \) samples per period, such that \( ML = 2^n \) points result in the processed record (fast FFT calculations). The number of data points is also chosen high enough to avoid aliasing, \( L > 2.5F \). The interpolated signal

\[
\hat{u}(q, T), \quad q = 0, \cdots, ML - 1
\]

is an estimate of \( z(qT/L) \). It is calculated starting from the measurements \( z(kh) \) using classical upsampling [29] and interpolation techniques [90]. The upsampling factor used is 6, an FIR filter of \( 48 \times 6 \) taps is used internally (designed with the Remez exchange procedure, pass band to \( 0.35f_s \), stopband from \( 0.6f_s \)). A cubic interpolation is made on the upsampled data. Using these choices, the systematic errors are 100 dB below the signal level up to \( 0.4f_s \).

3- Calculate the DFT spectrum (using the FFT)

\[
\hat{U}(s, T) = \frac{1}{ML} \sum_{q=0}^{ML-1} \hat{u}(q, T) e^{-j2\pi sq/L}.
\]  

\[ (10.9) \]

Note that the choice of \( L \) allows a fast calculation of the DFT using the FFT.

4- Define the cost function

\[
V(T) = \frac{\sum_{s=1}^{L} (|\hat{U}(Ms + 1, T)|^2 + |\hat{U}(Ms - 1, T)|^2)}{\sum_{s=1}^{L} |\hat{U}(Ms, T)|^2}.
\]

\[ (10.10) \]

This can be interpreted as the ratio of the power on the nonexcited frequency lines to that on the excited lines.
5- Define the estimate \( \hat{T} \) as
\[
\hat{T} = \arg \min_T V(T).
\] (10.11)

**Remark 10.3.** The idea behind the choice of the cost function in (10.10) is simple: if there is no leakage any more (the period fits perfectly), all the power of the signal should be at the multiples \( nT \). If there is leakage, a fraction of this power leaks to the neighboring lines. The cost function measures the ratio of this fraction and the algorithm minimizes it.

**Remark 10.4.** The minimization problem in (10.11) is nonlinear in \( T \). A nonlinear line search is used, that is initialized from \( \hat{T}_0 \). Since the cost function has many local minima, the search is split in a coarse search, scanning the cost function around the initial guess, followed by a fine search (based on parabolic interpolation) to eventually get the final estimate.

### 10.3.3 Estimation of the Fourier Coefficients and their Variance

Once an estimate \( \hat{T} \) is available, the full record is resampled according to this period length and split in \( M \) subrecords. For each subrecord the DFT spectrum is calculated. The final estimates of the Fourier coefficients and their variance are then obtained as the sample mean and sample variance of the spectra of these subrecords.

**Remark 10.5.** It can be noted that the sensitivity of the algorithm to the noise is very low and seems to be almost the same as that of the classical DFT (FFT) approach where the sample frequency would be known a priori. So, almost no price is paid for the fact that the basic period had to be extracted from the data instead of being given a priori.

### 10.3.4 Estimation of \( x_1(t) \), \( x_2(t) \) and \( \dot{x}_2(t) \)

Once the fundamental Fourier coefficients of the periodic signal are estimated, the estimates of the noise free periodic signal, \( \hat{x}_1(t) \), and its derivatives, \( \hat{x}_2(t) \) and \( \dot{\hat{x}}_2(t) \), are evaluated as (cf. Eq. (10.6))
\[
\hat{x}_1(t) = \sum_{r=-F}^{F} \hat{Z}_r e^{j2\pi rt/\hat{T}},
\]
\[
\hat{x}_2(t) = \sum_{r=-F}^{F} j \frac{2\pi r}{\hat{T}} \hat{Z}_r e^{j2\pi rt/\hat{T}},
\]
\[
\dot{\hat{x}}_2(t) = \sum_{r=-F}^{F} \left( j \frac{2\pi r}{\hat{T}} \right)^2 \hat{Z}_r e^{j2\pi rt/\hat{T}}.
\] (10.12)

**Remark 10.6.** Note that the LS-ASA algorithm is not more efficient than the LS algorithm due to the internal upsampling. This is because the cutoff frequency of the filter used is at \( 0.4f_s \), so almost the complete frequency band passes. The reason why the LS-ASA is more efficient is twofold:
• Only one frequency line in $M$ lines is used, the others are put to zeros ($M$ periods measured). This leads to a reduction of the noise power by $M$.

• Only the significant Fourier coefficients are used in the estimates $\hat{x}_1(t)$, $\hat{x}_2(t)$ and $\hat{\dot{x}}_2(t)$. Hence, an enormous noise reduction is obtained.

10.4 Simulation Study

The Van der Pol oscillator [62] was selected as the underlying system in this example. The oscillator is described by

$$
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{pmatrix} =
\begin{pmatrix}
x_2 \\
-x_1 + \varepsilon (1 - x_1^2) x_2
\end{pmatrix}.
$$

(10.13)

The Matlab routine \texttt{ode45} was used to solve (10.13) for $\varepsilon = 2$. The initial state of (10.13) was selected as $\begin{pmatrix} x_1(0) & x_2(0) \end{pmatrix}^T = (0 \ 1)^T$. All results below are based on data runs of length $N = 10^4$ with a sampling period $h = 0.1$ s. The measured signal $z(t)$ selected as the first state with white noise added. The differentiated signals $\hat{x}_2(t)$ and $\hat{\dot{x}}_2(t)$ for the LS estimation algorithm were obtained using first order difference (Euler center approximation) with $\hat{x}_1(t) = z(t)$. On the other hand, these signals were evaluated using (10.12) for the LS-ASA estimation algorithm.

In this example the estimated model used second-degree polynomials ($L = M = 2$). Both the LS and the LS-ASA algorithms were run for different SNRs. As a measure of performance,

$$
V = \frac{\|\hat{\theta}_N - \theta^*\|_2}{\|\theta^*\|_2}
$$

(10.14)
10.5 Conclusions

Figure 10.2: True (solid) and estimated (dashed) phase plots. LS estimate (left) and LS-ASA estimate (right) [SNR=20 dB].

was computed and plotted as a function of the SNR in Fig. 10.1. In (10.14), \( \theta^o \) denotes the true parameter vector. The phase plots for the estimated models at SNR of 20 dB and 60 dB in addition to the true system are given in Figures 10.2-10.3. Also, the estimated periodic signals at SNR of 20 dB and 60 dB compared to the true signal are given in Figures 10.4-10.5. The LS algorithm using the Euler center approximation did not give stable limit cycles for SNRs below 20 dB. The LS-ASA algorithm still gives good models for low SNRs as shown in Figures 10.6-10.7.

It can be concluded from Figures 10.1-10.7 that the LS-ASA algorithm gives significantly better estimates than the LS algorithm using finite difference approximation. Also, the LS-ASA algorithm gives more accurate estimates than the Markov estimation algorithm of Chapter 7, see Fig. (7.8). Moreover, the computational complexity of the LS-ASA algorithm is much lower than the Markov estimation algorithm and the MLE algorithm of Chapter 9.

10.5 Conclusions

A least squares estimation algorithm based on fully automated spectral analysis (ASA) technique has been introduced for the modeling of periodic signals using a second-order nonlinear ODE model. The ASA technique reduces the noise contributions on the modeled signal and its derivatives significantly. The suggested algorithm results in a much better performance as compared to the LS estimation algorithm using finite difference approximation. Also, the LS-ASA algorithm is computationally less demanding as compared to the Markov estimation algorithm and the MLE algorithm.
10. Periodic Signal Modeling Based on Fully Automated Spectral Analysis

Figure 10.3: True (solid) and estimated (dashed) phase plots. LS estimate (left) and LS-ASA estimate (right) [SNR=60 dB].

Figure 10.4: True (solid) and estimated (dashed) signals. LS estimate (left) and LS-ASA estimate (right) [SNR=20 dB].
Figure 10.5: True (solid) and estimated (dashed) signals. LS estimate (left) and LS-ASA estimate (right) [SNR=60 dB].

Figure 10.6: True (solid) and estimated LS-ASA (dashed) phase plots. 0 dB (left) and 10 dB (right).
Figure 10.7: True (solid) and estimated LS-ASA (dashed) signals. 0 dB (left) and 10 dB (right).
Chapter 11

Periodic Signal Modeling Based on Liénard’s Equation

11.1 Introduction

In the early days of nonlinear dynamics, it was found that many oscillating circuits can be modeled by the following second-order differential equation known as Liénard’s equation, see [125, 128],

\[ \ddot{y} + f(y)\dot{y} + g(y) = 0. \]  \hspace{1cm} (11.1)

Liénard’s equation can be interpreted mechanically as the equation of motion for a unit mass subject to a nonlinear damping force \(-f(y)\dot{y}\) and a nonlinear restoring force \(-g(y)\). Applications of Liénard’s equation can be found in many important examples. Examples include chemical reactions, growth of a single species, predator-prey systems and vibration analysis, see [74].

Choosing the state variables as \(x_1 = y(t)\) and \(x_2 = \dot{y}(t)\), Liénard’s equation is equivalent to the system

\[ \dot{x}_1 = x_2, \]
\[ \dot{x}_2 = -g(x_1) - f(x_1)x_2, \]  \hspace{1cm} (11.2)

which is known as the Liénard’s system.

In this chapter Liénard’s equation is used to model periodic signals following the approach introduced in Chapter 5. The conditions that guarantee the existence of periodic orbits for Liénard systems, see [57, 104], are used to reduce the number of parameters required to model periodic signals. This is expected to give significantly better parameter accuracy as compared to the approach used in Chapter 5 in case the modeled signal fulfills Liénard’s equation. Another advantage is that it can easily be checked if the estimated model will generate a stable periodic orbit.

The chapter is organized as follows. Section 11.2 introduces the details on the model based on Liénard’s equation. Section 11.3 analyzes the conditions imposed on the model to achieve the reduction in the parameters to be estimated. Section 11.4 presents a comparative simulation study between the approach taken in this chapter and the approach of Chapter 5. Conclusions appear in Section 11.5.
11.2 The ODE Model Based on Liénard’s Equation

11.2.1 Model Structures

The work done in Chapter 5 is based on modeling the signal \( y(t) \) by means of an unknown parameter vector \( \tilde{\theta} \) and an ODE of order two, i.e.

\[
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{pmatrix} = \begin{pmatrix} x_2(t) \\ f(x_1(t), x_2(t), \tilde{\theta}) \end{pmatrix},
\]

\[ y(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}. \tag{11.3}\]

The right hand side of the second state equation of (11.3) was expanded in terms of a truncated superposition of polynomial basis. Hence \( f(x_1(t), x_2(t), \tilde{\theta}) \) was parameterized as

\[
f(x_1(t), x_2(t), \tilde{\theta}) = \sum_{l_1=0}^{L_1} \sum_{m=0}^{M} \tilde{\theta}_{l_1,m} x_1^{l_1}(t)x_2^{m}(t),
\]

\[\tilde{\theta} = \left( \tilde{\theta}_{0,0}, \tilde{\theta}_{0,1}, \ldots, \tilde{\theta}_{0,M}, \tilde{\theta}_{1,0}, \ldots, \tilde{\theta}_{1,M}, \ldots, \tilde{\theta}_{L,0}, \ldots, \tilde{\theta}_{L,M} \right)^T. \tag{11.4}\]

In this chapter a Liénard model (11.2) is used for modeling periodic signals. Then the state space model (11.3) becomes

\[
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{pmatrix} = \begin{pmatrix} x_2(t) \\ -g(x_1(t), \theta_1) - f(x_1(t), \theta_2) x_2(t) \end{pmatrix},
\]

\[ y(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}. \tag{11.5}\]

\[
\theta = \left( \theta_1^T \quad \theta_2^T \right)^T. \tag{11.6}\]

Also here \( g(x_1(t), \theta_1) \) and \( f(x_1(t), \theta_2) \) are parameterized using (scalar) polynomial models

\[
g(x_1(t), \theta_1) = \sum_{l_1=1}^{L_1} \theta_{1,l_1} x_1^{l_1}(t),
\]

\[
f(x_1(t), \theta_2) = \sum_{l_2=0}^{L_2} \theta_{2,l_2} x_1^{l_2}(t). \tag{11.7}\]

Needless to say, the model (11.5) can be seen as a special case of the general case (11.3).

11.2.2 Discretization

Similarly to Chapter 5, the continuous time ODE model (11.5) is discretized in order to formulate complete discrete models. This is done here by exploiting
11.2. The ODE Model Based on Liénard’s Equation

an Euler forward numerical integration scheme. Selecting the discretization interval to be equal to the sampling period $h$, results in

$$
\begin{align*}
x_1(kh + h) &= x_1(kh) + hx_2(kh), \\
x_2(kh + h) &= x_2(kh) - h \sum_{l_1=1}^{L_1} \theta_{1,l_1} x_1^{l_1}(kh) - h \left( \sum_{l_2=0}^{L_2} \theta_{2,l_2} x_1^2(kh) \right) x_2(kh).
\end{align*}
$$

(11.8)

**Remark 11.1.** In the following, for notational convenience the dependence on $h$ is omitted assuming that $h$ is equal to one time unit. This means that an integer $k$ can be used as the time variable.

The model (11.8) can then be compactly written in the linear regression form as

$$
\begin{align*}
x_1(k + 1) - x_1(k) &= x_2(k), \\
x_2(k + 1) - x_2(k) &= -\phi^T (x_1(k), x_2(k)) \theta,
\end{align*}
$$

(11.9)

where

$$
\phi^T (x_1(k), x_2(k)) = \begin{pmatrix} x_1(k) & \cdots & x_1^{L_1}(k) & x_2(k) & \cdots & x_1^2(k)x_2(k) \end{pmatrix}, \\
\theta = \begin{pmatrix} \theta_{1,1} & \cdots & \theta_{1,L_1} & \theta_{2,0} & \cdots & \theta_{2,L_2} \end{pmatrix}^T.
$$

(11.10)

For comparison, discretizing the second state equation of (11.3) results in

$$
\begin{align*}
x_2(k + 1) - x_2(k) &= \tilde{\phi}^T (x_1(k), x_2(k)) \tilde{\theta},
\end{align*}
$$

(11.11)

where

$$
\begin{align*}
\tilde{\phi}^T (x_1(k), x_2(k)) &=
\begin{pmatrix}
1 & x_2(k) & \cdots & x_2^M(k) & x_1(k) & \cdots & x_1(k)x_2^M(k) & \cdots & x_1^2(k) & \cdots & x_1^2(k)x_2^M(k)
\end{pmatrix},
\end{align*}
$$

(11.12)

and the parameter vector $\tilde{\theta}$ is given by (11.4).

### 11.2.3 Algorithms

In order to derive different estimation schemes based on the model (11.9) there are at least three choices. The first choice is to formulate the model in a linear regression form using the measured data (cf. Section 5.3) and the approximations $\hat{x}_1(k) = z(k)$ and $\hat{x}_2(k) = z(k + 1) - z(k)$. Once the model is formulated in the linear regression form, a number of different estimation algorithms based on the least squares estimate, the Markov estimate (cf. Chapter 7), the Kalman filter (cf. Chapter 8) and the maximum likelihood estimate (cf. Chapter 9) can be developed from the model (11.9).

Another choice is to estimate the states $x_1(k)$ and $x_2(k)$ in addition to the parameter vector $\theta$ using the EKF as done in Chapter 8. In this case the
regression vector \( \phi^T(x_1(k), x_2(k)) \) is built up from the estimated states \( \hat{x}_1(k) \) and \( \hat{x}_2(k) \) rather than directly from measured data.

The third choice is to use algorithms based on the automated spectral analysis technique of Chapter 10 which finds more accurate estimates for the noise free periodic signal, \( x_1(k) \), and its time derivatives, \( x_2(k) \) and \( \dot{x}_2(k) \).

### 11.3 Model Parameterization Using Liénard’s Theorem

In this section the model assumptions introduced in Liénard’s theorem [57, 104] are considered and exploited. The theorem reads as follows.

**Theorem 11.1.** (Liénard’s theorem) Suppose that \( f(y) \) and \( g(y) \) satisfy the following conditions:

1. **Assumption 11.1.** \( f(y) \) and \( g(y) \) are continuously differentiable for all \( y \);
2. **Assumption 11.2.** \( g(-y) = -g(y) \) \( \forall \ y \) (i.e. \( g(y) \) is an odd function);
3. **Assumption 11.3.** \( g(y) > 0 \) for \( y > 0 \);
4. **Assumption 11.4.** \( f(-y) = f(y) \) \( \forall \ y \) (i.e. \( f(y) \) is an even function);
5. **Assumption 11.5.** The odd function \( F(y) = \int_y^0 f(u)du \) has exactly one positive zero at \( y = a \), is negative for \( 0 < y < a \), is positive and nondecreasing for \( y > a \), and \( F(y) \to \infty \) as \( y \to \infty \).

Then the system (11.2) has a unique, stable limit cycle surrounding the origin in the phase plane.

**Proof.** See [57]. \( \square \)

**Remark 11.2.** The assumptions on \( g(y) \) mean that the restoring force acts like an ordinary spring and tends to reduce any displacement. Also the assumptions on \( f(y) \) imply that the damping is negative at small \( |y| \) and positive at large \( |y| \). Since small oscillations are amplified and large oscillations are damped, the system tends to settle into a self-sustained oscillation of some intermediate amplitude.

Applying Assumption 11.2 and Assumption 11.4, the models (11.7) take the form

\[
\begin{align*}
g(x_1(t), \bar{\theta}_1) &= \sum_{l_1=0}^{L_1} \bar{\theta}_{1,l_1} x_1^{2l_1+1}(t), \\
f(x_1(t), \bar{\theta}_2) &= \sum_{l_2=0}^{L_2} \bar{\theta}_{2,l_2} x_1^{2l_2}(t), \\
\bar{\theta} &= \begin{pmatrix} \bar{\theta}_1^T \\ \bar{\theta}_2^T \end{pmatrix}^T = ( \bar{\theta}_{1,0} \ldots \bar{\theta}_{1,L_1} \bar{\theta}_{2,0} \ldots \bar{\theta}_{2,L_2})^T,
\end{align*}
\]
11.3. Model Parameterization Using Liénard’s Theorem

\[ L_1 = \frac{L_1 - 1}{2}, \quad L_2 = \frac{L_2}{2}. \]  

(11.15)

Hence the second state equation of (11.9) becomes

\[ x_2(k+1) - x_2(k) = -\phi^T(x_1(k), x_2(k)) \theta, \]  

(11.16)

where

\[ \phi^T(x_1(k), x_2(k)) = \begin{pmatrix} x_1(k) & x_1^3(k) & \cdots & x_1^{2L_1+1}(k) & x_2(k) & x_2^3(k)x_2(k) & \cdots & x_2^{2L_2}(k)x_2(k) \end{pmatrix}, \]  

(11.17)

and the parameter vector \( \theta \) is given by (11.14).

**Remark 11.3.** The reduction in the number of parameters has a great importance in two cases. First, when the modeled signal is highly corrupted with noise, i.e. the signal to noise ratio (SNR) is small, more inaccurate models are expected as the number of parameters increases. This is shown later in the numerical examples. A second case is when the modeled signal fullfills Liénard’s equation with high polynomial orders. Hence, the dimension of the parameter vector using general approaches will be high and this is expected to reduce the accuracy of the estimated model significantly. It may also lead to convergence problem for the EKF estimation algorithm of Chapter 8.

**Example 11.1.** Consider the following model with \( L_1 = 0 \) and \( L_2 = 1 \), i.e.

\[ g(x_1(t), \bar{\theta}_1) = \bar{\theta}_{1,0} x_1(t), \]
\[ f(x_1(t), \bar{\theta}_2) = \bar{\theta}_{2,0} + \bar{\theta}_{2,1} x_1^2(t). \]

(11.18)

It is clear that Assumptions 11.1-11.2 and Assumption 11.4 are satisfied. Assumption 11.3 gives \( \bar{\theta}_{1,0} > 0 \). Also Assumption 11.5 gives

\[ F(x_1) = x_1(t) \left( \bar{\theta}_{2,0} + \frac{1}{3} \bar{\theta}_{2,1} x_1^2(t) \right). \]

(11.19)

Thus \( a = \sqrt{-\frac{3\bar{\theta}_{2,0}}{\bar{\theta}_{2,1}}} \). Straightforward calculations show that \( F(x_1) \) is nondecreasing for \( x_1 > a \) provided that \( \bar{\theta}_{2,0} < 0 \). Also from Assumption 11.5

\[ F(x_1) < 0 \quad \forall \quad 0 < x_1 < \sqrt{-\frac{3\bar{\theta}_{2,0}}{\bar{\theta}_{2,1}}}, \quad \text{and} \quad F(x_1) > 0 \quad \forall \quad x_1 > \sqrt{-\frac{3\bar{\theta}_{2,0}}{\bar{\theta}_{2,1}}} \]

whenever \( \bar{\theta}_{2,0} < 0 \) and \( \bar{\theta}_{2,1} > 0 \). Finally \( F(x_1) \to \infty \) as \( x_1 \to \infty \) since for high values of \( x_1(t) \), \( F(x_1) \approx \frac{1}{3} \bar{\theta}_{2,1} x_1^3(t) \). Thus to guarantee the existence of a unique, stable limit cycle for the model (11.18), the parameters must satisfy \( \bar{\theta}_{1,0} > 0 \), \( \bar{\theta}_{2,0} < 0 \) and \( \bar{\theta}_{2,1} > 0 \).  

Next the more general case for the parameterization (11.13) is considered. The following theorem gives necessary conditions on \( g(x_1(t), \bar{\theta}_1) \) and \( f(x_1(t), \bar{\theta}_2) \) to guarantee the existence of a unique, stable limit cycle for general orders.
Theorem 11.2. Assume that the Liénard’s system is given by
\[
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{pmatrix} = \begin{pmatrix}
x_2(t) \\
-g(x_1(t), \bar{\theta}_1) - f(x_1(t), \bar{\theta}_2)
\end{pmatrix},
\]
\[y(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\
x_2(t) \end{pmatrix},\]
(11.20)
where the odd function \(g(x_1, \bar{\theta}_1)\) and the even function \(f(x_1, \bar{\theta}_2)\) are continuously differentiable polynomials given by
\[
g(x_1, \bar{\theta}_1) = \bar{\theta}_1, 0 + \bar{\theta}_1, 1 x_1 + \bar{\theta}_1, 2 x_1^2 + \cdots + \bar{\theta}_1, L_1 x_1^{L_1},
\]
\[
f(x_1, \bar{\theta}_2) = \bar{\theta}_2, 0 + \bar{\theta}_2, 1 x_1^2 + \cdots + \bar{\theta}_2, L_2 x_1^{L_2},
\]
(11.21)
Then this system has a unique, stable limit cycle encircling the origin in the phase plane if
Assumption 11.6. All zeros of the polynomial
\[A(s) = \bar{\theta}_1, 0 + \bar{\theta}_1, 1 s + \bar{\theta}_1, 2 s^2 + \cdots + \bar{\theta}_1, L_1 s^{L_1}\]
are in the LHP.
Assumption 11.7. The polynomial
\[B(s) = \bar{\theta}_2, 0 + \frac{\bar{\theta}_2, 1}{3} s + \frac{\bar{\theta}_2, 2}{5} s^2 + \cdots + \frac{\bar{\theta}_2, L_2}{2L_2 + 1} s^{L_2}\]
(11.23)
has exactly one positive real zero (say at \(a^2\)) and \(\bar{\theta}_2, 0 \leq 0\).
Assumption 11.8. \(f(x_1, \bar{\theta}_2) \geq 0\ \forall \ x_1 > a\).

Proof. See Appendix 11.A \qed

Remark 11.4. From Routh’s stability criterion for continuous systems, see [80], all zeros of \(A(s)\) are in the LHP if there are no sign changes in the left-most column of the Routh array. A necessary but not sufficient condition for this to happen is \(\bar{\theta}_1, i > 0\) for \(i = 0, \cdots, L_1\). Also from Descarte’s rule of sign [58], the number of positive real roots of \(B(s)\) is either equal to the number of variations of sign between successive terms in \(B(s)\) when arranged in descending powers of \(s\) or less than that number by an even integer. Thus \(B(s)\) should have an odd number of sign variations.

In addition to the reduction achieved in the number of estimated parameters, another advantage of the approach used in this chapter can be concluded. Since Theorem 11.2 gives more specific conditions on the parameters, these conditions can be used as a detection method for the existence of a unique, stable periodic orbit that models the periodic signal. Once the parameter vector \(\bar{\theta}\) is estimated, the polynomials \(A(s)\) and \(B(s)\) can be constructed and the conditions of Theorem 11.2 can be examined.

Remark 11.5. Note that there may be unique periodic solutions even in cases where the parameter constraints are not fulfilled. This is because the conditions of Theorem 11.1 are only sufficient for general periodic signals.
11.4 Numerical Examples

In this section a comparative simulation study for the two modeling approaches described by (11.3) and (11.5) is presented.

Example 11.2. In this example the data were generated using the following Liénard’s system

\[
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{pmatrix} =
\begin{pmatrix}
x_2 \\
-x_1 + (1 - 2x_1^4)x_2
\end{pmatrix}
\]  

(11.24)

which satisfies the conditions of Theorem 11.1. The Matlab routine \texttt{ode45} was used to solve (11.24). The initial states of (11.24) were selected as \((x_1(0) \ x_2(0))^T = (1 \ 0)^T\). All results below are based on data runs of length \(N = 3 \times 10^4\) samples with a sampling period \(h = 0.01\) s. The period of the solution of (11.24) was approximately 7 seconds. The measured signal was selected as the first state with white Gaussian noise added to obtain data with a SNR of 30 dB.

The model (11.3) was compared with the model (11.5) by modeling the periodic signal generated by (11.24) using the EKF estimation algorithm introduced in Chapter 8. The two EKF algorithms were initialized with

\[
\begin{pmatrix}
\hat{x}_1(0) \\
\hat{x}_2(0) \\
\hat{\theta}(0)
\end{pmatrix}^T = (-0.5 \ 0.5 \ 0)^T,
\]

\[
\begin{pmatrix}
\hat{x}_1(0) \\
\hat{x}_2(0) \\
\hat{\theta}(0)
\end{pmatrix}^T = (-0.5 \ 0.5 \ 0)^T.
\]

The remaining parameters were selected as \(P(0) = 10I, R_1 = 0.001I\) and \(r_2 = 1\). The orders of the models were chosen as \(L = 4, M = 1, L_1 = 0\) and \(L_2 = 2\). The number of estimated parameters was \((L + 1)(M + 1) = 10\) for the model (11.3) and \(L_1 + L_2 + 2 = 4\) for the model (11.5).

After \(3 \times 10^4\) samples the parameter estimates were as follows:

\[
\hat{\theta}^T = (0.005, 1.021, -1.149, -0.009, 0.008, -0.235, 0.086, 0.020, -0.011, -1.908)
\]

\[
\tilde{\theta}^T = (1.036, -0.990, 0.040, 2.102).
\]

Note the negative sign difference between \(\hat{\theta}\) and \(\tilde{\theta}\), cf. Eq. (11.11) and Eq. (11.16). Comparing the estimated parameter vector \(\tilde{\theta}\) with the results of Theorem 11.2 gives \(\tilde{\theta}_{1,0} = 1.036 > 0, \tilde{\theta}_{2,0} = -0.99 < 0\) and \(\tilde{\theta}_{2,2} = 2.102\). Thus \(A(s) = 1.036\) and \(B(s) = -0.99 + \frac{2.102}{2} s^2\). It is clear that \(A(s)\) does not have zeros in the RHP and \(B(s)\) has one positive zero at 1.535. Also \(f(x_1, \tilde{\theta}_2) = 2.102x_1^2 - 0.99 > 0\ \forall x_1 > \sqrt{1.535}\). This shows that the estimated model represents a unique, stable periodic orbit surrounding the origin in the phase plane.

The parameter estimates and phase plots for the system (11.24) and the estimated models are shown in Figures 11.1-11.2. The results indicate that
Figure 11.1: Parameter convergence.

Figure 11.2: True (solid) and estimated (dashed) phase plane plots for system (11.24). The model (11.5) left and the model (11.3) right. The models are initialized as $x_1(0) = -0.5$, $x_2(0) = 0.5$ and the final parameter vector estimate $\hat{\theta}$ or $\tilde{\theta}$, respectively, is applied.
11.4. Numerical Examples

Figure 11.3: True (solid) and estimated (dashed) phase plane plots for system (11.25). The model (11.5) left and the model (11.3) right. The models are initialized as $x_1(0) = -0.5$, $x_2(0) = 0.5$ and the final parameter vector estimate $\hat{\theta}$ or $\tilde{\theta}$, respectively, is applied.

the reduction in the number of estimated parameters achieved using models based on Liénard’s systems significantly increases the accuracy of the estimated models.

Example 11.3. Consider the following system

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ -x_1 + (1 - 3x_1^2 - 2x_2^2)x_2 \end{pmatrix}$$

(11.25)

which does not match Liénard’s system description. Similarly as done in Example 11.2, data length of $3 \times 10^4$ samples was generated from (11.25) with $h = 0.01$ s. The period of the solution of (11.25) was approximately 6 seconds. Hence, about 50 periods of the signal were measured with approximately 600 samples per period.

The model (11.3) was compared with the model (11.5) by modeling the periodic signal generated by (11.25) using the EKF estimation algorithm. The two EKF algorithms were initialized as in Example 11.2 except that $P(0) = 10^4 I$. The orders of the models were chosen as $L = 2$, $M = 3$, $\bar{L}_1 = 0$ and $\bar{L}_2 = 1$. The number of estimated parameters was 12 for the model (11.3) and 3 for the model (11.5).

The phase plots for the system (11.25) and the estimated models are shown in Fig. 11.3. The results show that even if the modeled signal does not fulfill Liénard’s equation, the underparameterized model (11.5) still represents a unique and stable periodic orbit that slightly deviates from the true periodic orbit and from the estimated periodic orbit using the model (11.3) with the correct polynomial orders.

Example 11.4. In this example 100 Monte-Carlo simulations were performed on the system (11.24) to study the performance of the two modeling approaches.
described by (11.3) and (11.5) compared to the Cramér-Rao bound (CRB) derived in Chapter 9 with different noise realizations. The data were generated as in Example 11.2. The two EKF algorithms were initialized with $P(0) = 10^{-4} I$, $R_1 = 10^{-6} I$, $r_2 = 1$ and

$$\left(\hat{x}_1(0) \quad \hat{x}_2(0) \quad \tilde{\theta}^T(0)\right)^T = \left(-0.5 \quad 0.5 \quad \bar{\theta}_o + \tilde{\sigma}(\text{SNR}, N)\right)^T,$$

$$\left(\hat{x}_1(0) \quad \hat{x}_2(0) \quad \tilde{\theta}^T(0)\right)^T = \left(-0.5 \quad 0.5 \quad \bar{\theta}_o + \bar{\sigma}(\text{SNR}, N)\right)^T,$$

where $\tilde{\theta}_o$ and $\bar{\theta}_o$ are the true parameter vectors, $\tilde{\sigma}(\text{SNR}, N)$ and $\bar{\sigma}(\text{SNR}, N)$ are the standard deviations as predicted by the CRB evaluated at specific SNR and data length $N$. The orders of the models were chosen as in Example 11.2.

The mean square error (MSE) and the CRB were evaluated for $N = 3 \times 10^4$ samples with different SNRs. The results are shown in Figures 11.4(a)-11.4(c). Also as a measure of performance,

$$V_1 = \frac{1}{n} \sum_{i=1}^{n} \|\tilde{\theta}_N - \tilde{\theta}_o\|_2,$$

$$V_2 = \frac{1}{n} \sum_{i=1}^{n} \|\bar{\theta}_N - \bar{\theta}_o\|_2$$

were computed and plotted as a function of the SNR in Fig. 11.4(d) ($n$ is the number of experiments in which convergence to the true parameter vector is achieved). Also, the Monte-Carlo experiments were repeated for different data lengths with SNR=10 dB. The initial values of $P$ and $R_1$ were chosen as $P(0) = 10^2 I$, $R_1 = 10^{-5} I$. The results are plotted in Fig. 11.5. The results of Figures 11.4-11.5 show that the approach of this chapter gives significantly better parameter estimates compared to the approach described by (11.3) especially for small SNR.

**Example 11.5.** In this example the suggested approach of this chapter is used to model a piece of a bell sound extracted from a CD in .wav format with a sampling frequency of 22.05 kHz. The least squares algorithm of Chapter 7 was applied to model 200 samples of the acoustic signal, see Fig 11.6(a). The model (11.5) with different combinations of $L_1$ and $L_2$ was used. It was noticed that the parameters of $\theta_2$ were very small compared to the parameters of $\theta_1$. Hence, $\theta_2$ was excluded from the model (11.5). Finally, the following model was obtained:

$$\left(\begin{array}{c} \dot{x}_1 \\ \dot{x}_2 \end{array}\right) = \left(\begin{array}{c} x_2 \\ -7.22 \times 10^7 x_1 + 1.14 \times 10^9 x_1^3 \end{array}\right).$$

The real data, the model output, the true and the estimated phase plots are given in Fig. 11.6. Note that the model obtained does not satisfy the conditions of Theorem 11.2. Specifically, the polynomial $A(s)$ has a zero in the RHP (at $s = 0.06$). Hence, there is a possibility that multi limit cycles could exist. Therefore, the model was initialized at $(x_1(0) \quad x_2(0))^T = (-0.08 \quad 0)^T$, i.e. close to the true limit cycle.
(a) MSE ($\tilde{\theta}_{1,0}$ (dashed), $\bar{\theta}_{1,0}$ (solid)) and
CRB ($\tilde{\theta}_{1,0}$ (dot-o), $\bar{\theta}_{1,0}$ (dot-x))

(b) MSE ($\tilde{\theta}_{0,1}$ (dashed), $\bar{\theta}_{2,0}$ (solid)) and
CRB ($\tilde{\theta}_{0,1}$ (dot-o), $\bar{\theta}_{2,0}$ (dot-x))

(c) MSE ($\tilde{\theta}_{4,1}$ (dashed), $\bar{\theta}_{2,2}$ (solid)) and
CRB ($\tilde{\theta}_{4,1}$ (dot-o), $\bar{\theta}_{2,2}$ (dot-x))

(d) $V_{1}$ (dashed) and $V_{2}$ (solid)

Figure 11.4: Statistical results vs. signal to noise ratio.
(a) MSE ($\hat{\theta}_{1,0}$ (dashed), $\bar{\theta}_{1,0}$ (solid)) and CRB ($\hat{\theta}_{1,0}$ (dot-o), $\bar{\theta}_{1,0}$ (dot-x))
(b) MSE ($\hat{\theta}_{0,1}$ (dashed), $\bar{\theta}_{2,0}$ (solid)) and CRB ($\hat{\theta}_{0,1}$ (dot-o), $\bar{\theta}_{2,0}$ (dot-x))
(c) MSE ($\hat{\theta}_{4,1}$ (dashed), $\bar{\theta}_{2,2}$ (solid)) and CRB ($\hat{\theta}_{4,1}$ (dot-o), $\bar{\theta}_{2,2}$ (dot-x))
(d) $V_1$ (dashed) and $V_2$ (solid)

Figure 11.5: Statistical results vs. data length.
11.5 Conclusions

The modeling of periodic signals using Liénard’s equation has been studied. The approach of this chapter is based on using the appropriate assumptions imposed in Liénard’s theorem to guarantee the existence of a unique and stable periodic orbit in the phase plane surrounding the origin. Using these conditions leads to a reduction in the ODE model parameters. This reduction not only reduces the computational load but also significantly increases the accuracy of the model in case the periodic signal does fulfill Liénard’s equation description compared to more general approaches. Moreover, the approach can be used for modeling signals that do not fulfill Liénard’s equation with only a small degradation in the accuracy of the estimated models.

11.A Proof of Theorem 11.2

First note that Assumptions 11.1-11.2 and Assumption 11.4 are true by construction. It remains to verify Assumption 11.3 and Assumption 11.5. Beginning with Assumption 11.3, the odd function $g(x_1, \bar{\theta}_1)$ can be written as

$$g(x_1, \bar{\theta}_1) = x_1 A(x_1^2).$$

Choosing $s = x_1^2$ gives Eq. (11.22). Further

$$g(x_1) > 0 \forall x_1 > 0 \text{ if } A(x_1^2) > 0.$$ 

The latter is the case if all zeros of $A(s)$ are in the LHP, since for negative zeros $A(s)$ takes the form

$$A(s) = \prod_{i=1}^{L_1} (s + z_i) \text{ for } z_i > 0,$$
where \(-z_i, i = 1, \cdots, \bar{L}_1\) are the zeros of \(A(s)\). Thus Assumption 11.6 implies Assumption 11.3.

**Remark 11.6.** Note that in case \(z_i\) is complex, i.e. \(z_i = z_{ir} + jz_{ic}\) and its complex conjugate is given as \(\bar{z}_i = z_{ir} - jz_{ic}\), it follows that
\[
(s + z_i)(s + \bar{z}_i) = s^2 + 2z_{ir}s + z_{ir}^2 + z_{ic}^2 > 0.
\]
This means that \(A(x_1^2) > 0\) for complex roots and Assumption 11.6 still implies Assumption 11.3.

Proceeding with Assumption 11.5, since \(f(x_1, \bar{\theta}_2)\) is an even function
\[
F(x_1) = x_1 \ B(x_1^2).
\]
To satisfy Assumption 11.5, \(B(x_1^2)\) should have exactly one positive zero at \(x_1 = a\), be negative for \(0 < x_1 < a\), be positive and nondecreasing for \(x_1 > a\), and \(F(x_1) \to \infty\) as \(x_1 \to \infty\). This means \(\bar{\theta}_{2,0}\) should be negative (for \(F(x_1)\) to be negative for small \(x_1\)) and \(B(s)\) given by (11.23) should have exactly one positive real zero. The nondecreasing condition on \(F(x_1)\) for \(x_1 > a\) means
\[
\frac{dF}{dx_1} \geq 0 \quad \text{for} \quad x_1 > a.
\]
Since
\[
F(x_1) = \int_{0}^{x_1} f(u)du,
\]
differentiating both sides gives
\[
\frac{dF}{dx_1} = f(x_1, \bar{\theta}_2).
\]
Thus \(F(x_1)\) is a nondecreasing function if Assumption 11.8 satisfied. Hence Assumption 11.7 and Assumption 11.8 together imply Assumption 11.5.
Chapter 12

Bias Analysis in LS Estimation of Periodic Signals

12.1 Introduction

As mentioned in Chapter 7, the Least squares (LS) estimator gives biased estimates at low signal to noise ratios (SNRs). The major reasons for this bias are the noise contamination of the derivatives of the modeled signal evaluated using Euler approximations, as well as discretization errors. In Chapter 10 the automated spectral analysis (ASA) technique was used to find more accurate estimates for the noise free periodic signal, \( x_1(t) \), and its time derivatives, \( x_2(t) \) and \( \dot{x}_2(t) \). Although the approach of Chapter 10 shows a significant improvement in the performance compared to using finite difference approximations, it is still interesting to study the bias in the LS estimation algorithm of Chapter 7 which is based on finite difference approximations.

The LS algorithm of Chapter 7 uses the linear regression equation

\[
\hat{x}_2(t) = \phi^T(\hat{x}_1(t), \hat{x}_2(t)) \theta + \varepsilon(t) \tag{12.1}
\]

for estimating the parameter vector \( \theta \). Here

\[
\phi^T(\hat{x}_1(t), \hat{x}_2(t)) = (1 \cdots \hat{x}_2^M(t) \cdots \hat{x}_1^L(t) \cdots \hat{x}_2^L(t) \hat{x}_2^M(t)), \tag{12.2}
\]

\[
\theta = (\theta_{0,0} \cdots \theta_{0,M} \cdots \theta_{L,0} \cdots \theta_{L,M})^T. \tag{12.3}
\]

The estimate \( \hat{x}_1(kh) \) of \( x_1(kh) \) is chosen to be equal to the modeled data (cf. Eq. (5.12)), i.e.

\[
\hat{x}_1(t) = z(t) = y(t) + e(t), \tag{12.4}
\]

where \( y(t) \) is the continuous time periodic signal and \( e(t) \) is a zero-mean white Gaussian noise. Also, \( \hat{x}_2(t) \) and \( \dot{x}_2(t) \) are the estimates of the first and second derivatives of the modeled signal, respectively, evaluated using finite difference approximation, and \( \varepsilon(t) \) is a regression error.

In this chapter, an analysis of the bias in the LS estimate of two periodic nonlinear systems is given. The goal is to study the effect of the sampling period, the signal to noise ratio and the system parameters on the estimation bias.

The chapter is organized as follows. Different estimation errors are discussed in Section 12.2. In Section 12.3, bias analysis and discretization errors...
for some simple periodic systems are evaluated. Section 12.4 gives a comparative numerical study between three different Euler approximation techniques. Conclusions appear in Section 12.5.

12.2 Estimation Errors

In order to estimate the parameter vector $\theta$ from the linear regression equation (12.1), some estimates of the modeled signal $x_1(kh)$, its first derivative $x_2(kh)$ and second derivative $\dot{x}_2(kh)$ are needed. Estimating these quantities using finite difference approximations from the measured data leads to some estimation errors. It is expected that the LS estimates will suffer from two estimation errors, namely: random noise errors and discretization errors. Random noise errors result due to differentiating additive measurement noise. On the other hand, discretization errors are caused by approximating the signal derivatives using finite difference schemes.

To investigate how the estimate $\hat{\theta}_N$ behaves when the data length $N$ becomes large, consider the following:

$$R = \mathbb{E} \left[ \phi(\hat{x}_1(t), \hat{x}_2(t)) \, \phi^T(\hat{x}_1(t), \hat{x}_2(t)) \right],$$  \hspace{1cm} (12.5)

$$r = \mathbb{E} \left[ \phi(\hat{x}_1(t), \hat{x}_2(t)) \, \dot{\hat{x}}_2(t) \right],$$  \hspace{1cm} (12.6)

where

$$\phi^T(\hat{x}_1(t), \hat{x}_2(t)) = \left( 1 \cdots \hat{x}_2^M(t) \cdots \hat{x}_1^L(t) \cdots \hat{x}_1^L(t) \hat{x}_2^M(t) \right),$$  \hspace{1cm} (12.7)

and

$$\mathbb{E}_f(t) = \lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} \mathbb{E}_f(t).$$  \hspace{1cm} (12.8)

Remark 12.1. $\mathbb{E}$ is used instead of the ordinary expectation $\mathbb{E}$ to account for noise-free signals. For fully random signals $\mathbb{E} = \mathbb{E}$.

In this case the asymptotic parameter vector estimate $\hat{\theta}$ is given by

$$\hat{\theta} = R^{-1} \, r = \theta_0 + \hat{\theta}_b,$$  \hspace{1cm} (12.9)

where $\theta_0$ is the true parameter vector and $\hat{\theta}_b$ is the bias vector. Similarly

$$R = R_0 + \hat{R}_b,$$  \hspace{1cm} (12.10)

$$r = r_0 + \hat{r}_b,$$  \hspace{1cm} (12.11)

where

$$R_0 = \mathbb{E} \left[ \phi(x_1(t), x_2(t)) \, \phi^T(x_1(t), x_2(t)) \right],$$  \hspace{1cm} (12.12)

$$r_0 = \mathbb{E} \left[ \phi(x_1(t), x_2(t)) \, \dot{x}_2(t) \right].$$  \hspace{1cm} (12.12)
and $\tilde{R}_b$ and $\tilde{r}_b$ are the bias contributions to $R_0$ and $r_0$, respectively, due to using estimated states $\hat{x}_1(t)$ and $\hat{x}_2(t)$ instead of the true states.

Now using (12.9)-(12.11) gives

$$\bar{\theta} = (R_0 + \tilde{R}_b)^{-1}(r_0 + \tilde{r}_b)$$

$$= R_0^{-1}r_0 + (R_0 + \tilde{R}_b)^{-1}(\tilde{r}_b - \tilde{R}_b\theta_0).$$

(12.13)

**Remark 12.2.** The bias vector $\tilde{\theta}_b$ is a contribution of random noise errors and discretization errors. The contributions of random noise errors and discretization errors are denoted in this chapter by $\tilde{\theta}_n$ and $\tilde{\theta}_d$, respectively. Hence

$$\tilde{\theta}_b = \tilde{\theta}_n + \tilde{\theta}_d,$$

(12.14)

where

$$\tilde{\theta}_n = (R_0 + \tilde{R}_n)^{-1}(\tilde{r}_n - \tilde{R}_n\theta_0),$$

(12.15)

$$\tilde{\theta}_d = (R_0 + \tilde{R}_d)^{-1}(\tilde{r}_d - \tilde{R}_d\theta_0).$$

(12.16)

Also, here $\tilde{R}_n$ and $\tilde{r}_n$ are the bias contributions to $R_0$ and $r_0$, respectively, due to random noise errors; $R_d$ and $\tilde{r}_d$ are the bias contributions to $R_0$ and $r_0$, respectively, due to discretization errors.

The bias vector $\tilde{\theta}_b$ will depend on the sampling period $h$ and the derivative approximations. Derivative approximations can always be chosen by the user. A general approximation for the derivatives of the modeled signal is [97]

$$D^n f(t) = \frac{1}{h^n} \sum_j \Gamma_{n,j} f(t + jh),$$

(12.17)

which has to satisfy the following conditions

$$\sum_j \Gamma_{n,j} j^v = \begin{cases} 0 & v = 0, \ldots, n - 1 \\ n! & v = n. \end{cases}$$

(12.18)

The frequency function of the differentiator (12.17) then satisfies

$$D^n(e^{j\omega}) = (j\omega)^n + O(|\omega|^{n+1}),$$

(12.19)

where $O(\cdot)$ denotes the ordo concept. This means that the low-frequency asymptote of the differentiator frequency function is $(j\omega)^n$. See [97, 98] for more details.

**Remark 12.3.** As can be noticed from (12.18), the minimal number of terms in (12.17) is $n + 1$. In such a case $D^n$ will be a high-pass filter. If the number of coefficients is increased, the gained degree of freedom can be used to decrease the high frequency gain to avoid differentiating the noise as much as possible. See [25] for more details about design of differentiating filters.
In this chapter the estimation of the parameter vector will be considered for the following three simple finite difference approximations of $x_2(kh)$ and $\dot{x}_2(kh)$ which are special cases from (12.17) (in all approximations, $\hat{x}_1(kh) = z(kh)$ was chosen):

- **A1**: Euler backward approx. (EB)

  \[
  \tilde{x}_2(kh) = \frac{1}{h} \sum_{j=-1}^{0} \Gamma_{1,j} \tilde{x}_1(kh + jh), \quad \Gamma_{1,-1} = -1 \quad \text{and} \quad \Gamma_{1,0} = 1,
  \]

  \[
  \hat{x}_2(kh) = \frac{1}{h^2} \sum_{j=-2}^{0} \Gamma_{2,j} \tilde{x}_1(kh + jh), \quad \Gamma_{2,-2} = 1, \quad \Gamma_{2,-1} = -2 \quad \text{and} \quad \Gamma_{2,0} = 1.
  \]  

  (12.20)

- **A2**: Euler forward approx. (EF)

  \[
  \tilde{x}_2(kh) = \frac{1}{h} \sum_{j=0}^{1} \Gamma_{1,j} \tilde{x}_1(kh + jh), \quad \Gamma_{1,0} = -1 \quad \text{and} \quad \Gamma_{1,1} = 1,
  \]

  \[
  \hat{x}_2(kh) = \frac{1}{h^2} \sum_{j=0}^{2} \Gamma_{2,j} \tilde{x}_1(kh + jh), \quad \Gamma_{2,0} = 1, \quad \Gamma_{2,1} = -2 \quad \text{and} \quad \Gamma_{2,2} = 1.
  \]  

  (12.21)

- **A3**: Euler center approx. (EC)

  \[
  \tilde{x}_2(kh) = \frac{1}{h} \sum_{j=-1}^{1} \Gamma_{1,j} \tilde{x}_1(kh + jh), \quad \Gamma_{1,-1} = -\frac{1}{2}, \quad \Gamma_{1,0} = 0 \quad \text{and} \quad \Gamma_{1,1} = \frac{1}{2},
  \]

  \[
  \hat{x}_2(kh) = \frac{1}{h^2} \sum_{j=-2}^{2} \Gamma_{2,j} \tilde{x}_1(kh + jh), \quad \Gamma_{2,-2} = \frac{1}{4}, \quad \Gamma_{2,-1} = 0, \quad \Gamma_{2,0} = -\frac{1}{2}, \quad \Gamma_{2,1} = 0 \quad \text{and} \quad \Gamma_{2,2} = \frac{1}{4}.
  \]  

  (12.22)

**Remark 12.4.** **A1-A3** are chosen as examples for finite difference approximation. Many other different approximations can be considered. For example, one can think about evaluating $\tilde{x}_2(kh)$ and $\hat{x}_2(kh)$ using two different finite difference approximations.

It is well known from the numerical analysis literature that EC approximation gives lower discretization error compared to EB and EF approximations. This is because EC approximation gives a higher order (in $h$) remainder term when a Taylor series expansion is performed. Hence the EC approximation provides smaller remainder term (for small $h$) than the EB and EF approximations. Also, the discretization error in this case is expected to decrease as $h$ decreases. On the other hand, random noise error gives both bias and variance
contributions to the LS estimates. It is expected that the noise error increases as \( h \) decreases since the noise will be highly amplified (\( h \) appears in the denominator of the differentiator). Small values of \( h \) will only be suitable in the case the SNR is high. The effect of \( h \) on the noise errors can be somewhat lower if more general derivative approximations such as (12.17) are used.

In the next section some simple systems will be considered to analyze different estimation errors. The first aim of this study is to deduce if the bias in the LS estimates will follow the same expectations as for discretization errors and random noise errors. The second aim is to check if these two errors are additive so we can find an optimal sampling period for each system.

### 12.3 Explicit Analysis for Simple Systems

In this section the study will concentrate on the following two nonlinear systems:

\[ S_1: \quad \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ -\eta x_1^3 \end{pmatrix}. \quad (12.23) \]

\[ S_2: \quad \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ \alpha x_1 + \beta x_2 + \gamma x_1^3 x_2 \end{pmatrix}. \quad (12.24) \]

**Remark 12.5.** Note that \( S_1 \) does not have a unique stable periodic orbit as is the case in \( S_2 \). Therefore, the amplitude of the periodic signal generated by \( S_1 \) is fully determined by the initial state, see [62].

#### 12.3.1 Random noise errors

The random noise errors can be calculated by exploiting the superposition principle as follows. Let \( \tilde{x}_1(kh) = x_1(kh) + \tilde{x}_1(kh) \) where \( \tilde{x}_1(kh) \) represents the noise contribution. It is clear from Eqs. (5.17) and (12.4) that \( \tilde{x}_1(kh) = e(kh) \). Similarly, \( \tilde{x}_2(kh) \) and \( \tilde{x}_2(kh) \) for Eqs. (12.20)-(12.22) can be evaluated. Therefore, the noise contribution in \( \tilde{x}_2(kh) \) and \( \tilde{x}_2(kh) \) are as follows:

- **A1**: (EB)
  \[
  \tilde{x}_2(kh) = e(kh) - e(kh - h), \\
  \tilde{\dot{x}}_2(kh) = \frac{e(kh) - 2e(kh - h) + e(kh - 2h)}{h^2}. \quad (12.25)
  \]

- **A2**: (EF)
  \[
  \tilde{x}_2(kh) = \frac{e(kh + h) - e(kh)}{h}, \\
  \tilde{\dot{x}}_2(kh) = \frac{e(kh + 2h) - 2e(kh + h) + e(kh)}{h^2}. \quad (12.26)
  \]
• A3: (EC)

\[
\tilde{x}_2(kh) = \frac{e(kh + h) - e(kh - h)}{2h}, \\
\tilde{\dot{x}}_2(kh) = \frac{e(kh + 2h) - 2e(kh) + e(kh - 2h)}{4h^2}.
\] (12.27)

Remark 12.6. Note that by Assumption 5.2, \( \mathbb{E}(\tilde{x}_i^1) = 0, \forall i \) odd, \( \mathbb{E}(\tilde{x}_i^2) = \sigma^2 \), and \( \mathbb{E}(\tilde{x}_i^4) = 3\sigma^4 \), and \( \mathbb{E}(\tilde{x}_i^6) = 15\sigma^6 \).

Remark 12.7. Note that for noise-free periodic data

\[
\mathbb{E}(x_1^1 x_2) = \frac{1}{T} \int_0^T x_1^1(t)x_2(t)dt = \frac{1}{T} \int_0^T x_1^1(t)\dot{x}_1(t)dt
\]

\[
= \frac{1}{T(i + 1)} [x_1^{i+1}(T) - x_1^{i+1}(0)] = 0, \forall i.
\] (12.28)

Similarly, \( \mathbb{E}(x_2^1 \dot{x}_2) = 0, \forall i. \)

In the following two examples, the bias contribution due to random noise errors is analyzed for \( S_1 \) and \( S_2 \).

Example 12.1. Random noise errors of \( S_1 \).

Here \( \phi = -\tilde{x}_1^1 \). Thus \( R = \mathbb{E}(\tilde{x}_1^2) \) and \( r = \mathbb{E}(\tilde{\dot{x}}_1^2) \). Straightforward calculations give

\[
R = \mathbb{E}(x_1^1 + \tilde{x}_1^1)^6
\]

\[
= \underbrace{\mathbb{E}(x_1^6)}_{R_0} + 15\sigma^2\underbrace{\mathbb{E}(x_1^4)}_{\hat{R}_0} + 45\sigma^4\underbrace{\mathbb{E}(x_1^2)}_{R_n} + 15\sigma^6.
\] (12.29)

Similarly,

\[
r = \mathbb{E}[-(x_1^1 + \tilde{x}_1^1)^3(\dot{x}_2 + \dot{\tilde{x}}_2)]
\]

\[
= -\mathbb{E}(x_1^1 \dot{x}_2) - 3\mathbb{E}(x_1^1)\mathbb{E}(\tilde{x}_1 \dot{x}_2) - 3\sigma^2\mathbb{E}(x_1 \dot{x}_2) - \mathbb{E}(\tilde{x}_1^3 \dot{x}_2)
\]

\[
= \eta\underbrace{\mathbb{E}(x_1^6)}_{R_0} - 3\mathbb{E}(x_1^4)\mathbb{E}(\tilde{x}_1 \dot{x}_2) + 3\sigma^2\eta\mathbb{E}(x_1^2) - \mathbb{E}(\tilde{x}_1^3 \dot{x}_2).
\] (12.30)

Since \( R \) and \( r \) are scalars in this case, Eq. (12.15) gives

\[
\theta = \frac{r}{R_0} + \frac{\tilde{r}_n R_0 - \sigma^0 \hat{R}_n}{\hat{R}_0 R}.
\] (12.31)

Thus it follows that

\[
\tilde{\theta}_n = \frac{\tilde{r}_n - \eta \hat{R}_n}{R}.
\] (12.32)
12.3. Explicit Analysis for Simple Systems

Table 12.1: Values of some expectations.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>EB</th>
<th>EF</th>
<th>EC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{E}(\tilde{x}_2^2)$</td>
<td>$2\sigma^2/h^2$</td>
<td>$2\sigma^2/h^2$</td>
<td>$\sigma^2/2h^2$</td>
</tr>
<tr>
<td>$\mathbb{E}(\tilde{x}_1\tilde{x}_2)$</td>
<td>$\sigma^2/h$</td>
<td>$-\sigma^2/h$</td>
<td>$0$</td>
</tr>
<tr>
<td>$\mathbb{E}(\tilde{x}_1^2\tilde{x}_2^2)$</td>
<td>$4\sigma^4/h^2$</td>
<td>$4\sigma^4/h^2$</td>
<td>$\sigma^4/2h^2$</td>
</tr>
<tr>
<td>$\mathbb{E}(\tilde{x}_1\tilde{x}_2)$</td>
<td>$3\sigma^4/h$</td>
<td>$-3\sigma^4/h$</td>
<td>$0$</td>
</tr>
<tr>
<td>$\mathbb{E}(\tilde{x}_1^2)$</td>
<td>$18\sigma^6/h^2$</td>
<td>$18\sigma^6/h^2$</td>
<td>$3\sigma^6/2h^2$</td>
</tr>
<tr>
<td>$\mathbb{E}(\tilde{x}_2^2)$</td>
<td>$3\sigma^2/h^3$</td>
<td>$-3\sigma^2/h^3$</td>
<td>$0$</td>
</tr>
<tr>
<td>$\mathbb{E}(\tilde{x}_1\tilde{x}_2)$</td>
<td>$\sigma^2/h^2$</td>
<td>$\sigma^2/h^2$</td>
<td>$-\sigma^2/2h^2$</td>
</tr>
<tr>
<td>$\mathbb{E}(\tilde{x}_1^2\tilde{x}_2)$</td>
<td>$3\sigma^4/h^2$</td>
<td>$3\sigma^4/h^2$</td>
<td>$-3\sigma^4/2h^2$</td>
</tr>
<tr>
<td>$\mathbb{E}(\tilde{x}_1^2\tilde{x}_2)$</td>
<td>$5\sigma^4/h^3$</td>
<td>$-5\sigma^4/h^3$</td>
<td>$0$</td>
</tr>
</tbody>
</table>

Equation (12.32) shows that an unbiased estimate of $\eta$ can be obtained if both $r_0$ and $R_0$ are perturbed proportionally to the ratio between them, i.e. $r_0/R_0 = \tilde{r}/R_n$.

Now, the bias $\tilde{\theta}_n$ can be evaluated for $A1$-$A3$ replacing $\mathbb{E}(\tilde{x}_1\tilde{x}_2)$ and $\mathbb{E}(\tilde{x}_1^2\tilde{x}_2)$ by their corresponding values as indicated in Table 12.1 which can be straightforwardly derived using Eqs. (12.35)-(12.37). Hence,

$$\tilde{\theta}_n^{EB} = \tilde{\theta}_n^{EF} = -\frac{3\sigma^2}{h^2} \mathbb{E}(x_1^2) - 12\sigma^2 h \mathbb{E}(x_1^4) - 3\sigma^4 \left( \frac{1}{h^2} + 5\sigma^2 \eta \right) \mathbb{E}(x_1^4) + 15\sigma^4 \mathbb{E}(x_1^2) + 15\sigma^6,$$  (12.33)

$$\tilde{\theta}_n^{EC} = -\frac{3\sigma^2}{h^2} \mathbb{E}(x_1^2) - 12\sigma^2 h \mathbb{E}(x_1^4) + 3\sigma^4 \left( \frac{1}{h^2} - 5\sigma^2 \eta \right) \mathbb{E}(x_1^4) + 15\sigma^2 \mathbb{E}(x_1^2) + 15\sigma^6.$$  (12.34)

For high SNR, it can be assumed that $\mathbb{E}(x_1^6) \gg 15\sigma^2 \mathbb{E}(x_1^4)$ and that the effect of the terms which contain $\sigma^4$ and $\sigma^6$ can be neglected. Hence,

$$\theta_n^{EB} = \theta_n^{EF} = -\frac{3\sigma^2}{h^2} \mathbb{E}(x_1^2) - 12\eta \mathbb{E}(x_1^4) + O(\sigma^4),$$  (12.35)

$$\theta_n^{EC} = -\frac{3\sigma^2}{h^2} \mathbb{E}(x_1^2) - 12\eta \mathbb{E}(x_1^4) + O(\sigma^4).$$  (12.36)

Also, multiplying the numerators of Eqs. (12.35)-(12.36) by $\mathbb{E}(x_1^2)/\mathbb{E}(x_1^4)$, and noticing that $\text{SNR} = \mathbb{E}(x_1^2)/\sigma^2$ gives

$$\theta_n^{EB} = \theta_n^{EF} \approx -\frac{3}{h^2} \frac{\mathbb{E}(x_1^2)^2}{\mathbb{E}(x_1^4)} - 12\eta \mathbb{E}(x_1^2) \mathbb{E}(x_1^4),$$  (12.37)

$$\theta_n^{EC} \approx -\frac{3}{h^2} \frac{\mathbb{E}(x_1^2)^2}{\mathbb{E}(x_1^4)} - 12\eta \mathbb{E}(x_1^2) \mathbb{E}(x_1^4).$$  (12.38)
Therefore, $\tilde{\theta}_n$ satisfies the following relations for high SNR and for small $h$:

$$\tilde{\theta}_n \propto \frac{1}{\text{SNR}}, \quad (12.39)$$

$$\tilde{\theta}_n \propto \frac{1}{h^2}. \quad (12.40)$$

**Example 12.2.** Random noise errors of $\mathcal{S}2$. In this case, $\phi^T = (\tilde{x}_2, \tilde{x}_1, \tilde{x}_2^T\tilde{x}_2)$ and $\theta = (\beta, \alpha, \gamma)^T$. Therefore

$$R = \mathbb{E}(\phi\phi^T) = \mathbb{E}\left(\begin{array}{c}
\tilde{x}_2^2 \\
\tilde{x}_1\tilde{x}_2 \\
\tilde{x}_2\tilde{x}_2^T \tilde{x}_2
\end{array}\right), \quad (12.41)$$

and

$$r = \mathbb{E}(\phi \tilde{x}_2) = \mathbb{E}\left(\begin{array}{c}
\tilde{x}_2\tilde{x}_2 \\
\tilde{x}_1\tilde{x}_2 \\
\tilde{x}_2^T \tilde{x}_2 \tilde{x}_2
\end{array}\right). \quad (12.42)$$

Similarly as done in Example 12.1, replacing $\tilde{x}_1$, $\tilde{x}_2$ and $\tilde{x}_2$ by $(x_1 + \tilde{x}_1)$, $(x_2 + \tilde{x}_2)$ and $(\tilde{x}_2 + \tilde{x}_2)$, it follows that

$$R(1, 1) = \mathbb{E}(x_2^2) + \mathbb{E}(\tilde{x}_2^2), \quad (12.43)$$

$$R(1, 2) = \mathbb{E}(\tilde{x}_1\tilde{x}_2), \quad (12.44)$$

$$R(1, 3) = \mathbb{E}(x_1^2x_2^2) + \sigma^2\mathbb{E}(x_2^2) + \mathbb{E}(x_1^n\tilde{x}_2^2) + \mathbb{E}(\tilde{x}_1^2\tilde{x}_2^2), \quad (12.45)$$

$$R(2, 2) = \mathbb{E}(x_1^2) + \sigma^2, \quad (12.46)$$

$$R(2, 3) = 3\mathbb{E}(x_1^2)\mathbb{E}(\tilde{x}_1\tilde{x}_2) + \mathbb{E}(x_1^n\tilde{x}_2^2), \quad (12.47)$$

$$R(3, 3) = \mathbb{E}(x_1^4x_2^2) + 6\sigma^4\mathbb{E}(x_1^n\tilde{x}_2^2) + 3\sigma^4\mathbb{E}(x_2^2)$$

$$+ \mathbb{E}(x_1^2)\mathbb{E}(\tilde{x}_2^2) + 6\mathbb{E}(x_1^2)\mathbb{E}(\tilde{x}_1\tilde{x}_2^2) + \mathbb{E}(\tilde{x}_1^2\tilde{x}_2^2), \quad (12.48)$$

$$r(1) = \mathbb{E}(\tilde{x}_2\tilde{x}_2), \quad (12.49)$$

$$r(2) = \alpha \mathbb{E}(x_1^2) + \mathbb{E}(\tilde{x}_1\tilde{x}_2), \quad (12.50)$$

$$r(3) = \beta \mathbb{E}(x_1^2) + \gamma \mathbb{E}(x_1^n\tilde{x}_2^2) + 2\alpha \mathbb{E}(x_1^2)\mathbb{E}(\tilde{x}_1\tilde{x}_2)$$

$$+ \mathbb{E}(x_1^2)\mathbb{E}(\tilde{x}_2\tilde{x}_2) + \mathbb{E}(\tilde{x}_1^2\tilde{x}_2\tilde{x}_2). \quad (12.51)$$

Hence, using Eq. (12.15) and Table 12.1, $\tilde{\theta}_n$ can be evaluated numerically for $A1$-$A3$.

Asymptotic expressions for the random noise bias on the parameters $\beta$, $\alpha$ and $\gamma$ (denoted as $\tilde{\beta}_n$, $\tilde{\alpha}_n$ and $\tilde{\gamma}_n$, respectively) can be evaluated using Eq. (12.15), Table 12.1, and Eqs. (12.43)-(12.51). Straightforward calculations assuming high SNR and small $h$ for $A3$, see Appendix 12.A, give

$$\tilde{\beta}_n \approx \frac{\mathbb{E}(x_1^2)}{2h^2\text{SNR}} \times$$

$$\frac{\beta \left( \mathbb{E}(x_1^2)\mathbb{E}(x_1^n\tilde{x}_2^2) - \mathbb{E}(x_1^n\tilde{x}_2^2) \right) + \gamma \left( \mathbb{E}(x_1^2)\mathbb{E}(x_1^2\tilde{x}_2^2) - \mathbb{E}(x_1^n)\mathbb{E}(x_1^2) \right)}{\left( \mathbb{E}(x_1^n)\mathbb{E}(x_1^n\tilde{x}_2^2) - [\mathbb{E}(x_1^n\tilde{x}_2^2)]^2 \right)}, \quad (12.52)$$
\[ \tilde{\alpha}_n \approx -\frac{1}{2h^2\text{SNR}}, \quad (12.53) \]
\[ \tilde{\gamma}_n \approx \frac{E(x_1^2)}{2h^2\text{SNR}} \times \left[ \beta \left( E(x_1^2x_2^2) - E(x_1^2)E(x_2^2) \right) + \gamma \left( E(x_1^2)E(x_1^2x_2^2) - E(x_1^2)E(x_2^2) \right) \right] \left( E(x_2^2)E(x_1^2x_2^2) - |E(x_1^2x_2^2)|^2 \right), \quad (12.54) \]

Also, for \( \beta = -\gamma = \varepsilon \), \( \|\tilde{\theta}_n\| \) can be approximated as
\[ \|\tilde{\theta}_n\| \approx \frac{\varepsilon}{2h^2\text{SNR}} \sqrt{L^2 + Q^2 + \frac{1}{\varepsilon^2}}, \quad (12.55) \]

where
\[ L = E(x_1^2) \left( E(x_1^2)E(x_1^2x_2^2) + E(x_1^2)E(x_1^2x_2^2) - E(x_1^2)E(x_1^2x_2^2) \right), \quad (12.56) \]
\[ Q = E(x_1^2) \left( E(x_1^2x_2^2) + E(x_1^2)E(x_2^2) - E(x_1^2)E(x_2^2) - E(x_1^2)E(x_1^2x_2^2) \right). \quad (12.57) \]

Hence
\[ \|\tilde{\theta}_n\| \propto \frac{1}{\text{SNR}}, \quad (12.58) \]
\[ \|\tilde{\theta}_n\| \propto \frac{1}{h^2}. \quad (12.59) \]

### 12.3.2 Discretization errors

In this section the evaluation of discretization errors is considered. The data are assumed to be noise-free (i.e. \( \hat{x}_1(kh) = x_1(kh) \)) and the estimates \( \hat{x}_2(kh) \) and \( \hat{\dot{x}}_2(kh) \) are chosen as one of \( A1-A3 \). The discretization error contributions to \( \hat{x}_2(kh) \) and \( \hat{\dot{x}}_2(kh) \) can be evaluated using a Taylor series expansion assuming:

**Assumption 12.1.** The solution to the ODE model described by Eq. (5.18) is sufficiently differentiable.

The discretization errors for \( A1-A3 \) can be summarized as follows, see Appendix 12.B:

- **A1:** (EB)
  \[ \begin{align*}
  \hat{x}_2(kh) &= -\frac{h}{2} D^2 x_1(kh) + \frac{h^2}{6} D^3 x_1(kh) - \frac{h^3}{24} D^4 x_1(kh) + O(h^4), \\
  \hat{\dot{x}}_2(kh) &= -h D^2 x_2(kh) + \frac{7h^2}{12} D^3 x_2(kh) + O(h^3).
  \end{align*} \quad (12.60) \]
12. Bias Analysis in LS Estimation of Periodic Signals

- A2: (EF)

\[ \tilde{x}_2(kh) = \frac{h}{2} D^2 x_1(kh) + \frac{h^2}{6} D^3 x_1(kh) + \frac{h^3}{24} D^4 x_1(kh) + O(h^4), \]

\[ \hat{x}_2(kh) = h D^2 x_2(kh) + \frac{7h^2}{12} D^3 x_2(kh) + O(h^3). \]  \hspace{1cm} (12.61)

- A3: (EC)

\[ \tilde{x}_2(kh) = \frac{h^2}{6} D^3 x_1(kh) + \frac{h^4}{120} D^5 x_1(kh) + O(h^6), \]

\[ \hat{x}_2(kh) = \frac{h^2}{3} D^3 x_2(kh) + O(h^4). \]  \hspace{1cm} (12.62)

In the next two examples, discretization errors of \( S_1 \) and \( S_2 \) are evaluated.

**Example 12.3. Discretization errors of \( S_1 \).**

Here \( \phi = -x_1^3 \). Thus \( R = E(x_1^q) \) and \( r = E(-x_1^3 \dot{x}_2) \). Equation (12.9) gives

\[ \bar{\theta} = \frac{E(-x_1^3 \dot{x}_2)}{E(x_1^q)} = \frac{E(-x_1^3 \dot{x}_2)}{E(x_1^q)} + \frac{E(-x_1^3 \dot{x}_2)}{\dot{\theta}_d}. \]  \hspace{1cm} (12.63)

Noticing that (cf. Remark 12.7)

\[ E(x_1^3 D^2 x_2) = E(x_1^3 D \ddot{x}_2) = E(x_1^3 D [-\eta x_1^3]) = -3\eta E(x_1^3 \dot{x}_1) = 0, \]  \hspace{1cm} (12.64)

and using Eqs. (12.60)-(12.62), it follows that

\[ \bar{\theta}_d^{EB} = \bar{\theta}_d^{EF} = -\frac{7h^2}{12} \frac{E(x_1^3 D^3 x_2)}{E(x_1^q)} + O(h^3), \]  \hspace{1cm} (12.65)

\[ \bar{\theta}_d^{EC} = -\frac{h^2}{3} \frac{E(x_1^3 D^3 x_2)}{E(x_1^q)} + O(h^4). \]  \hspace{1cm} (12.66)

It also holds that \( x_1^3 D^3 x_2 = -6\eta x_1^3 x_2^2 + 3\eta^2 x_1^5 \). Therefore Eqs. (12.65)-(12.66) can alternatively be written as

\[ \bar{\dot{\theta}}_d^{EB} = \dot{\theta}_d^{EF} = \frac{7h^2}{4} \frac{\eta E(2x_1^4 x_2^2 - \eta x_1^5)}{E(x_1^q)} + O(h^3), \]  \hspace{1cm} (12.67)

\[ \bar{\dot{\theta}}_d^{EC} = \frac{h^2}{3} \frac{\eta E(2x_1^4 x_2^2 - \eta x_1^5)}{E(x_1^q)} + O(h^4). \]  \hspace{1cm} (12.68)

The discretization errors of \( S_1 \) can be reduced to be of \( O(h^3) \) by considering the general differentiator form given in (12.17), see Appendix 12.C. Also, it is concluded from Eqs. (12.67)-(12.68) that

\[ \bar{\dot{\theta}}_d \propto h^2. \]  \hspace{1cm} (12.69)
12.3. Explicit Analysis for Simple Systems 187

Now the total estimation bias \( \tilde{\theta} \) can be found for \( S_1 \) by the addition of Eqs. (12.67)-(12.68) and Eqs. (12.37)-(12.38), respectively. Thus

\[
\tilde{\theta}_{EB} \approx 7h^2\eta \frac{E(2x_1^4x_2^2 - \eta x_1^8)}{E(x_1^4)} + \frac{3}{2\eta^2} \frac{E(x_1^8)^2 - 12\eta E(x_1^4)E(x_1^4)}{SNR E(x_1^4)},
\]

\[
\tilde{\theta}_{EC} \approx h^2\eta \frac{E(2x_1^4x_2^2 - \eta x_1^8)}{E(x_1^4)} + \frac{3}{2\eta^2} \frac{E(x_1^8)^2 - 12\eta E(x_1^4)E(x_1^4)}{SNR E(x_1^4)}.
\]

Differentiating \( |\tilde{\theta}|^2 \) using Eqs. (12.70)-(12.71) w.r.t. the sampling period \( h \) to find the optimal sampling period value that minimize the norm of the total bias \( \tilde{\theta} \) gives

\[
h_{opt}^{EB} = h_{opt}^{EF} \approx \left| \frac{6[E(x_1^4)]^2}{7\eta SNR E(2x_1^4x_2^2 - \eta x_1^8)} \right|^\frac{1}{4},
\]

\[
h_{opt}^{EC} \approx \left[ \frac{6E(x_1^4)E(x_1^4)}{SNR E(2x_1^4x_2^2 - \eta x_1^8)} \left( 1 - \sqrt{1 - \frac{SNR E(2x_1^4x_2^2 - \eta x_1^8)}{24\eta[E(x_1^4)]^2}} \right) \right]^\frac{1}{4}.
\]

Remark 12.8. Differentiating \( |\tilde{\theta}|^2 \) for \( S_1 \) means solving \( 2\tilde{\theta} \frac{d\tilde{\theta}}{dn} = 0 \). Since \( \tilde{\theta}^{EB} = 0 \) does not have any real roots, expression (12.72) was found by solving \( \tilde{\theta}^{EC} = 0 \) for positive real roots.

Example 12.4. Discretization errors of \( S_2 \).

In this case, \( \phi^T = (\hat{x}_2 \ x_1 \ x_1^2 \hat{x}_2) \). Therefore

\[
R = E(\phi\phi^T) = E \begin{pmatrix} \hat{x}_2^2 & x_1\hat{x}_2 & x_1^2\hat{x}_2^2 \\ x_1\hat{x}_2 & x_1^2 & x_1^2x_2^2 \\ x_1^2\hat{x}_2^2 & x_1^2x_2^2 & x_1^4x_2^4 \end{pmatrix},
\]

and

\[
r = E(\phi\hat{x}_2) = E \begin{pmatrix} \hat{x}_2 \ x_1\hat{x}_2 \ x_1^2\hat{x}_2 \ x_1^2x_2 \ x_1^2x_2^2 \ x_1^4x_2^4 \end{pmatrix}.
\]

Now using Eqs. (12.60)-(12.62), discretization errors on each component of \( R \) and \( r \) can be computed. The results are given in Table 12.2, see Appendix 12.D. Also, discretization errors can be evaluated as a function of \( \alpha, \beta \) and \( \gamma \) by using \( \dot{D}x_2 = \alpha x_1 + \beta x_2 + \gamma x_1^2x_2 \). The results in this case are given in Table 12.3, see Appendix 12.D. Finally \( \tilde{\theta}_e \) can be evaluated numerically for different Euler approximations using Eq. (12.16).
<table>
<thead>
<tr>
<th>Quantity</th>
<th>EB</th>
<th>EB</th>
<th>EB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{r}_d(1)$</td>
<td>$0$</td>
<td>$0$</td>
<td>$O(h^4)$</td>
</tr>
<tr>
<td>$\tilde{r}_d(2)$</td>
<td>$0$</td>
<td>$0$</td>
<td>$O(h^4)$</td>
</tr>
<tr>
<td>$\tilde{r}_d(3)$</td>
<td>$0$</td>
<td>$0$</td>
<td>$O(h^4)$</td>
</tr>
</tbody>
</table>

Table 12.2: Discretization errors of $S_2.$
### Table 12.3: Discretization errors of $S_2$ as a function of the system parameters.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>EB</th>
<th>EF</th>
<th>EC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{R}_d(1,1)$</td>
<td>$h^2 \left( \frac{\alpha}{2} + \frac{\beta^2 - \gamma}{12} \right) E(x_1^2) + \frac{\alpha}{2} E(x_1^2)$</td>
<td>$h^2 \left( \frac{\alpha}{2} + \frac{\beta^2 - \gamma}{12} \right) E(x_1^2) + \frac{\alpha}{2} E(x_1^2)$</td>
<td>$\frac{h^2}{3} \left[ (\alpha + \beta^2) E(x_1^2) + 2\beta \gamma E(x_1^2) \right] + 2\gamma E(x_1^2) + \gamma^2 E(x_1^2) + O(h^3)$</td>
</tr>
<tr>
<td>$\tilde{R}_d(1,2)$</td>
<td>$-\frac{h}{4} E(x_1) + O(h^2)$</td>
<td>$\frac{h}{4} E(x_1^2) + O(h^2)$</td>
<td>$\frac{h^2}{6} \left[ \alpha \beta E(x_1^2) + 2\beta \gamma E(x_1^2) + \alpha \gamma E(x_1^2) \right] + O(h^4)$</td>
</tr>
<tr>
<td>$\tilde{R}_d(1,3)$</td>
<td>$-h \left[ \beta E(x_1^2) + \gamma E(x_1^2) \right] + O(h^2)$</td>
<td>$h \left[ \beta E(x_1^2) + \gamma E(x_1^2) \right] + O(h^2)$</td>
<td>$\frac{h^2}{3} \left[ (\alpha + \beta^2) E(x_1^2) + 2\beta \gamma E(x_1^2) \right] + 2\gamma E(x_1^2) + \gamma^2 E(x_1^2) + O(h^4)$</td>
</tr>
<tr>
<td>$\tilde{R}_d(2,2)$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
</tr>
<tr>
<td>$\tilde{R}_d(3,3)$</td>
<td>$-h \left[ \beta E(x_1^2) + \gamma E(x_1^2) \right] + O(h^2)$</td>
<td>$h \left[ \beta E(x_1^2) + \gamma E(x_1^2) \right] + O(h^2)$</td>
<td>$\frac{h^2}{3} \left[ (\alpha + \beta^2) E(x_1^2) + 2\beta \gamma E(x_1^2) \right] + 2\gamma E(x_1^2) + \gamma^2 E(x_1^2) + O(h^4)$</td>
</tr>
<tr>
<td>$\tilde{R}_d(1)$</td>
<td>$-h \left[ \alpha E(x_1^2) + \beta \gamma E(x_1^2) + \gamma E(x_1^2) \right] + O(h^2)$</td>
<td>$h \left[ \alpha E(x_1^2) + \beta \gamma E(x_1^2) + \gamma E(x_1^2) \right] + O(h^2)$</td>
<td>$h^2 \left[ \frac{\alpha \beta}{12} E(x_1^2) + \gamma \left( \frac{\alpha \beta}{6} + \frac{\beta^2 \gamma}{3} \right) E(x_1^2) + \frac{\alpha \gamma}{2} E(x_1^2) \right] + \beta(\frac{\alpha \gamma}{2} + \frac{\beta \gamma^2}{2}) E(x_1^2) + 3\beta \gamma E(x_1^2) + \frac{3 \beta \gamma^2}{2} E(x_1^2) + 3 \gamma^2 E(x_1^2) + O(h^4)$</td>
</tr>
<tr>
<td>$\tilde{R}_d(2)$</td>
<td>$-h \left[ \alpha E(x_1^2) + \beta \gamma E(x_1^2) + \gamma E(x_1^2) \right] + O(h^2)$</td>
<td>$h \left[ \alpha E(x_1^2) + \beta \gamma E(x_1^2) + \gamma E(x_1^2) \right] + O(h^2)$</td>
<td>$\frac{h^2}{3} \left[ (\alpha + \beta^2) E(x_1^2) + 2\beta \gamma E(x_1^2) \right] + 2\gamma E(x_1^2) + \gamma^2 E(x_1^2) + O(h^4)$</td>
</tr>
<tr>
<td>$\tilde{R}_d(3)$</td>
<td>$-h \left[ \alpha E(x_1^2) + \beta \gamma E(x_1^2) + \gamma E(x_1^2) \right] + O(h^2)$</td>
<td>$h \left[ \alpha E(x_1^2) + \beta \gamma E(x_1^2) + \gamma E(x_1^2) \right] + O(h^2)$</td>
<td>$h^2 \left[ \frac{\alpha \beta}{12} E(x_1^2) + \gamma \left( \frac{\alpha \beta}{6} + \frac{\beta^2 \gamma}{3} \right) E(x_1^2) + \frac{\alpha \gamma}{2} E(x_1^2) \right] + \beta(\frac{\alpha \gamma}{2} + \frac{\beta \gamma^2}{2}) E(x_1^2) + 3\beta \gamma E(x_1^2) + \frac{3 \beta \gamma^2}{2} E(x_1^2) + 3 \gamma^2 E(x_1^2) + O(h^4)$</td>
</tr>
</tbody>
</table>
12. Bias Analysis in LS Estimation of Periodic Signals

In Appendix 12.E, asymptotic expressions for the discretization bias on the parameters $\beta$, $\alpha$ and $\gamma$ (denoted as $\tilde{\beta}_d$, $\tilde{\alpha}_d$ and $\tilde{\gamma}_d$, respectively) using $\mathcal{A}3$ are derived at high SNR and it is proved that

$$\|\tilde{\theta}_d\| \propto h^2. \quad (12.76)$$

Now using Eqs. (12.52)-(12.54) and Eqs. (12.129)-(12.131) the norm of the total estimation bias is given by

$$\|\tilde{\theta}\| \approx \sqrt{(\tilde{\beta}_n + \tilde{\beta}_d)^2 + (\tilde{\alpha}_n + \tilde{\alpha}_d)^2 + (\tilde{\gamma}_n + \tilde{\gamma}_d)^2}. \quad (12.77)$$

Differentiating $\|\tilde{\theta}\|^2$ w.r.t. the sampling period $h$ gives (for $\beta = -\gamma = \varepsilon$)

$$h_{opt}^{EC} \approx \frac{9}{(SNR)^2} \times \left| \frac{|R_0|^2}{|E(x_1^2)|^2} + \varepsilon^2 |E(x_1^2)|^2 (T_4^1 + T_5^2) \right|^2 + 4(T_2^1 + T_3^2 + T_7^1) + 4\varepsilon_T T_3 T_5 T_7, \quad (12.78)$$

where

$$T_1 = E(x_1^2)E(x_1^2 x_2^2) - E(x_1^2)E(x_1^2 x_2^2) - E(x_1^2)E(x_1^2 x_2^2) + E(x_1^2)E(x_1^2 x_2^2), \quad (12.79)$$

$$T_2 = E(x_1^2)E(x_2^2) - E(x_1^2)E(x_2^2) - E(x_1^2)E(x_2^2), \quad (12.80)$$

$$T_3 = E(x_1^2)E(x_1^2 x_2^2) - E(x_1^2)E(x_2^2) - E(x_1^2)E(x_2^2) - E(x_1^2)E(x_2^2) + E(x_1^2)E(x_2^2) - E(x_1^2)E(x_2^2) + E(x_1^2)E(x_2^2) + E(x_1^2)E(x_2^2), \quad (12.81)$$

$$T_4 = E(x_1^2 D^3 x_2) - \frac{1}{2}E(x_1^2 D^2 x_2) + \frac{1}{2}E(x_1^2 D^2 x_2), \quad (12.82)$$

$$T_5 = E(x_1^2 x_2^2) - E(x_1^2)E(x_2^2) - E(x_1^2)E(x_1^2 x_2^2) + E(x_1^2)E(x_2^2), \quad (12.83)$$

$$T_6 = E(x_1^2)E(x_1^2 x_2^2) - E(x_1^2)E(x_2^2) - E(x_1^2)E(x_2^2) + E(x_1^2)E(x_2^2), \quad (12.84)$$

$$T_7 = E(x_1^2)E(x_2^2) - E(x_1^2)E(x_2^2) - E(x_1^2)E(x_2^2) - E(x_1^2)E(x_2^2)$$

$$- E(x_1^2)E(x_1^2 x_2^2) - E(x_1^2)E(x_1^2 x_2^2), \quad (12.85)$$

Similarly, analytical expressions for $h_{opt}^{EB}$ and $h_{opt}^{EF}$ can be derived by repeating the analysis of Appendix 12.A and Appendix 12.E for $\mathcal{A}1$ and $\mathcal{A}2$, respectively.

12.4 Numerical Study

In this section a numerical study of the estimation errors of $S1$ and $S2$ using $\mathcal{A}1$-$\mathcal{A}3$ is given. The study is based on numerical evaluation of the truncated analytical expressions derived in Section 12.3. In the numerical calculation of the derived expressions, $E(f(t))$ was approximated by $\frac{1}{N} \sum_{t=1}^{N} f(t)$.
Example 12.5. Random noise error study.

In this example $10^4$ data samples were generated from $S_1$ and $S_2$ using the Matlab routine ode45 for $\eta = 2$, $\alpha = -1$, $\beta = 2$ and $\gamma = -2$. The initial states of $S_1$ and $S_2$ were selected as $(x_1(0) \ x_2(0))^T = (0 \ 1)^T$ and $(x_1(0) \ x_2(0))^T = (2 \ 0)^T$, respectively. These initial states were chosen so that transient effects can be neglected. First, the bias error for these systems was studied at different SNRs with a sampling period $h = 0.05$ s. Second, the bias was studied at different sampling periods. In this case, the SNR was 50 dB for $S_1$ and 60 dB for $S_2$. Finally, the bias was studied for different values of the system parameters. The results are shown in Figures 12.1-12.6.

As it is shown in the log-log scale Figures 12.1-12.2, $\log \|\hat{\theta}_n\|$ using $A_1-A_3$ is proportional to $(- \log(SNR) + \text{constant})$ for moderate and high SNR. Also, it
12. Bias Analysis in LS Estimation of Periodic Signals

Figure 12.3: $|\tilde{\theta}_n|$ vs. $h$ for $S1 [\eta = 2, \text{SNR} = 50 \text{ dB}]$.

Figure 12.4: $\|\tilde{\theta}_n\|$ vs. $h$ for $S2 [\alpha = -1, \beta = -\gamma = 2, \text{SNR} = 60 \text{ dB}]$.

can be concluded from the log-log scale Figures 12.3-12.4 that $\log \|\tilde{\theta}_n\|$ is proportional to $(-\log(h) + \text{constant})$. These results match the derived asymptotic results in Eqs. (12.39)-(12.40) and Eqs. (12.58)-(12.59), and the earlier expectation that random noise errors and its corresponding bias $\tilde{\theta}_n$ can be reduced by increasing the sampling period $h$ or using smaller $h$ whenever the SNR is high. Also, it can be noticed that A3 gives the lowest bias as expected.

Figures 12.5-12.6 show that the random noise bias increases as the value of $\eta$ in $S1$ and $\epsilon$ in $S2 (\beta = \epsilon, \gamma = -\epsilon)$ is increased. This is so because the nonlinear dynamics of the systems $S1$ and $S2$ become more effective as $\eta$ and $\epsilon$, respectively, increases. Hence, the signals generated by $S1$ and $S2$ become more nonlinearly distorted and the accuracy of the finite difference approximations decreases.
12.4. Numerical Study

Example 12.6. Discretization error study.

In this example, a noise-free data length of $10^4$ samples was generated from the systems $S_1$ and $S_2$ as done in Example 12.5. The bias corresponding to discretization errors ($\hat{\theta}_d$) was evaluated using Eqs. (12.67)-(12.68) for $S_1$, and Table 12.3 and Eq. (12.16) for $S_2$. The results are plotted in the log-log scale Figures 12.7-12.8.

The results of Figures 12.7-12.8 show that $\log \|\hat{\theta}_d\|$ is proportional to $(\log(h) + \text{constant})$, see Eqs. (12.69) and (12.76). This result also matches our earlier expectation that the discretization error increases as the sampling period is increased. It can be noticed also from Figures 12.7-12.8 that the slope of $\|\hat{\theta}_d^{EC}\|$ is lower than the slopes of $\|\hat{\theta}_d^{EB}\|$ and $\|\hat{\theta}_d^{EF}\|$, i.e. the EC approximation (A3) gives the lowest discretization bias.
Also, $\tilde{\theta}_d$ can be evaluated for different system parameters. The results are shown in Figures 12.9-12.10. These figures show that $\|\tilde{\theta}_d\|$ increases as the value of $\eta$ in $S_1$ and $\varepsilon$ in $S_2$ is increased. This behaviour can be explained here, as done in Example 12.5, by the accuracy reduction of the derivative approximations due to increasing the nonlinear dynamics of the two systems.

It can be concluded from Example 12.5 and Example 12.6 that there are two contradicting requirements to obtain an accurate estimate for the parameter vector $\theta$ using the least squares estimation algorithm. A small sampling period $h$ is needed to reduce the bias contribution due to discretization errors $\tilde{\theta}_d$, and a large $h$ is required to reduce the bias contribution due to random noise errors $\tilde{\theta}_n$. Therefore, it is expected that there is an optimal sampling period $h_{\text{opt}}$ that achieves minimum total estimation bias. This $h_{\text{opt}}$ can be easily determined by...
adding the two bias contributions and plotting the results versus \( h \) as shown in Figures 12.11-12.12. Figure 12.11 shows that \( h_{opt} = 0.052 \) s gives minimum estimation bias for \( S_1 \). The evaluation of the approximate analytical values of \( h_{opt}^{EB} \) (\( = h_{opt}^{EF} \)) and \( h_{opt}^{EC} \), see Eqs. (12.72)-(12.73), gives 0.0425 s and 0.0490 s, respectively. Figure 12.12 shows that \( h_{opt} = 0.03 \) s for \( S_2 \) in case EB or EF approximations are used. On the other hand, if EC approximation is used, the optimal sampling period is \( h_{opt} = 0.02 \) s. The expression of \( h_{opt}^{EC} \) in Eq. (12.78) gives 0.0190 s.

The difference between the value of \( h_{opt} \) concluded from Fig. 12.11 and the derived analytical expressions for \( S_1 \) is expected to be reduced as the SNR increases. This is due to the fact that the analytical expressions are derived for high SNR values. In order to verify this expectation, the total estimation bias was re-evaluated for SNR=70 dB. The results in this case are given in
12. Bias Analysis in LS Estimation of Periodic Signals

![Graph](image1)

**Figure 12.11:** $|\tilde{\theta}|$ vs. $h$ for $S1 [\eta = 2$, SNR=50 dB].

![Graph](image2)

**Figure 12.12:** $\|\tilde{\theta}\|$ vs. $h$ for $S2 [\alpha = -1, \beta = -\gamma = 2$, SNR=60 dB].

Fig. 12.13. Figure 12.13 shows that $h_{opt} = 0.016$ s. The evaluation of the approximate analytical values of $h_{opt}^{EB}$ and $h_{opt}^{EC}$ in this case gives 0.0133 s and 0.0153 s, respectively.

12.5 Conclusions

Estimation errors in the LS modeling of periodic signals using second-order nonlinear ODE have been studied in this chapter. The estimation of the parameters of two nonlinear systems were considered using different Euler approximation techniques. The analysis shows that the sampling period ($h$) plays a critical role in the performance of the LS estimation algorithm. Asymptotic analytical expressions show that

$$\|\tilde{\theta}_n\| \propto \frac{1}{\sqrt{SNR}}, \quad \|\tilde{\theta}_d\| \propto h^2$$
and that the estimation bias increases whenever the nonlinear dynamics become more effective. Hence, the existence of an optimal sampling period \( h_{opt} \) that achieves the lowest estimation bias is expected. It is shown in the chapter how an analytical expression for \( h_{opt} \) can be derived in a systematic way for different periodic nonlinear systems.

12.A Asymptotic Random Noise Bias for \( S^2 \) Using \( A^3 \)

Using Table 12.1 and Eqs. (12.43)-(12.51), it follows that (neglecting \( \sigma^4 \) and \( \sigma^6 \) terms)

\[
R(1,1) = \mathbb{E}(x_1^2) + \frac{\sigma^2}{2h^2},
\]

(12.86)

\[
R(1,2) = 0,
\]

(12.87)

\[
R(1,3) = \mathbb{E}(x_1^2x_2^2) + \sigma^2\mathbb{E}(x_2^2) + \frac{\sigma^2}{2h^2}\mathbb{E}(x_1^3),
\]

(12.88)

\[
R(2,2) = \mathbb{E}(x_1^2) + \sigma^2,
\]

(12.89)

\[
R(2,3) = 0,
\]

(12.90)

\[
R(3,3) = \mathbb{E}(x_1^4x_2^2) + 6\sigma^2\mathbb{E}(x_1^2x_2^2) + \frac{\sigma^2}{2h^2}\mathbb{E}(x_1^4),
\]

(12.91)

\[
r(1) = 0,
\]

(12.92)

\[
r(2) = \alpha\mathbb{E}(x_1^3) - \frac{\sigma^2}{2h^2},
\]

(12.93)

\[
r(3) = \beta\mathbb{E}(x_1^4x_2^2) + \gamma\mathbb{E}(x_1^2x_2^2).
\]

(12.94)

Hence

\[
\hat{r}_n = \sigma^2 \begin{pmatrix} 0 & -\frac{1}{\sigma^2} & 0 \end{pmatrix}^T + \mathcal{O}(\sigma^4),
\]

(12.95)
and

\[ \tilde{R}_n = \sigma^2 \begin{pmatrix} \frac{1}{2h^2} & 0 & \frac{1}{2h^2} \tilde{E}(x_1^2) + \frac{1}{2h^2} \tilde{E}(x_2^2) \\ 0 & 1 & 0 \\ \frac{1}{2h^2} \tilde{E}(x_1^2) + \frac{1}{2h^2} \tilde{E}(x_2^2) & 0 & 6\tilde{E}(x_1^2x_2^2) + \frac{1}{2h^2} \tilde{E}(x_1^4) \end{pmatrix} + \mathcal{O}(\sigma^4). \]  

12.B Discretization Error Contributions to $\hat{x}_2(kh)$ and $\hat{x}_2(kh)$

As an example, consider using the Euler backward approximation ($A_1$). Then

$$\hat{x}_2(kh) = \frac{x_1(kh) - x_1(kh - h)}{h}. \quad (12.105)$$

Performing a Taylor series expansion of $x_1(kh - h)$ around $x_1(kh)$, the result is

$$x_1(kh - h) = x_1(kh) - hDx_1(kh) + \frac{h^2}{2}D^2x_1(kh) - \frac{h^3}{6}D^3x_1(kh)$$

$$+ \frac{h^4}{24}D^4x_1(kh) + O(h^5). \quad (12.106)$$

Substituting by (12.106) in (12.105), it follows that

$$\hat{x}_2(kh) = \frac{Dx_1(kh) - \frac{h}{2}D^2x_1(kh) + \frac{h^2}{6}D^3x_1(kh) - \frac{h^3}{24}D^4x_1(kh) + O(h^4)}{r_2(kh)}. \quad (12.107)$$

Similarly,

$$\hat{x}_2(kh) = \frac{\hat{x}_2(kh) - \hat{x}_2(kh - h)}{h} = \frac{1}{h} \left[ x_2(kh) - x_2(kh - h) - \frac{h}{2}D^2x_1(kh) + \frac{h^2}{6}D^3x_1(kh) \right.\right.$$

$$\left. - \frac{h^3}{24}D^4x_1(kh) - \frac{h^3}{24}D^4x_1(kh) + O(h^4) \right]. \quad (12.108)$$

Thus performing the following Taylor series expansions

$$x_2(kh - h) = x_2(kh) - hDx_2(kh) + \frac{h^2}{2}D^2x_2(kh) - \frac{h^3}{6}D^3x_2(kh) + O(h^4),$$

$$D^2x_1(kh - h) = D^2x_1(kh) - hD^3x_1(kh) + \frac{h^2}{2}D^4x_1(kh) + O(h^3),$$

$$D^3x_1(kh - h) = D^3x_1(kh) - hD^4x_1(kh) + O(h^2),$$

$$D^4x_1(kh - h) = D^4x_1(kh) + O(h),$$

equation (12.108) becomes

$$\hat{x}_2(kh) = \frac{Dx_2(kh) - hD^2x_2(kh) + \frac{7h^2}{12}D^3x_2(kh) + O(h^3)}{\hat{x}_2(kh)}. \quad (12.109)$$

This completes the derivation of Eq. (12.60). Similarly, $\hat{x}_2(kh)$ and $\hat{x}_2(kh)$ can be evaluated for $A_2$ and $A_3$. 
12. Bias Analysis in LS Estimation of Periodic Signals

12.C Discretization Error of Order $O(h^3)$ for $S1$

For $S1$

$$E(x_1^6) \tilde{\eta} = -E(\hat{x}_1 x_1^3)$$

$$= -E \left[ \frac{1}{h^2} \sum_j \Gamma_{2,j} x_1 (kh + jh) \times x_1^3(kh) \right]$$

$$= -\frac{1}{h^2} \sum_j \Gamma_{2,j} E \left[ \sum_{\nu=0}^{\infty} \frac{1}{\nu!} (jh)^\nu D^\nu x_1 \times x_1^3 \right]$$

$$= -\frac{1}{h^2} \sum_{\nu=0}^{\infty} \frac{1}{\nu!} h^\nu \left( \sum_j \Gamma_{2,j} j^\nu \right) E \left( D^\nu x_1 \times x_1^3 \right).$$

(12.110)

Now using (12.18) and noticing that $E(x_1^3 D^3 x_1) = E(x_1^3 D^2 x_2) = 0$ from (12.64), straightforward calculations give

$$\tilde{\eta} = \eta - \frac{1}{E(x_1^6)} \sum_{\nu=4}^{\infty} \frac{1}{\nu!} h^{\nu-2} \left( \sum_j \Gamma_{2,j} j^\nu \right) E \left( D^\nu x_1 \times x_1^3 \right)$$

$$= \eta + \tilde{\eta}_2 h^2 + \tilde{\eta}_3 h^3 + \cdots$$

(12.111)

In order to have discretization errors of $O(h^3)$, $\tilde{\eta}_2$ should hence be equal to 0. Since

$$\tilde{\eta}_2 = -\frac{1}{4E(x_1^6)} \left( \sum_j \Gamma_{2,j} j^4 \right) E \left( D^4 x_1 \times x_1^3 \right),$$

(12.112)

$$\tilde{\eta}_2 = 0$$

if

$$\sum_j \Gamma_{2,j} j^4 = 0.$$

One can design the differentiation filter so that $\sum_j \Gamma_{2,j} j^4 = 0$. For example, consider using a filter with 5 coefficients based on EF and EB approximations and including low pass filter effects with unity static gain, i.e.

$$H(q) = \frac{(q - \rho)}{(1 - \rho)} \left( 1 - q^{-1} \right) \frac{(1 - q^{-1})}{(\rho - 1)}.$$  

(12.113)

where $\rho$ is a free parameter to be determined to achieve that $\sum_{j=-2}^{2} \Gamma_{2,j} j^4 = 0$. In this case it holds that

$$\Gamma_{2,0} = -\frac{2(1 + \rho + \rho^2)}{(1 - \rho)^2},$$

$$\Gamma_{2,1} = \Gamma_{2,-1} = \frac{(1 + \rho)^2}{(1 - \rho)^2},$$

$$\Gamma_{2,2} = \Gamma_{2,-2} = -\frac{\rho}{(1 - \rho)^2}.$$  

(12.114)
Thus
\[ \sum_{j=-2}^{2} \Gamma_{2,j} j^4 = \frac{1}{(1-\rho)^2} \left[ -2\rho(2)^4 + 2(1+\rho)^3(1)^4 - 2(1+\rho+\rho^2)(0)^3 \right] \]
\[ = \frac{2}{(1-\rho)^2} (\rho^2 - 14\rho + 1). \]
(12.115)

It follows that \( \sum_{j=-2}^{2} \Gamma_{2,j} j^4 = 0 \) for \( \rho = 7 \pm 4\sqrt{3} \). Therefore, choosing \( \rho = 7 - 4\sqrt{3} \) or \( \rho = 7 + 4\sqrt{3} \) gives
\[ \tilde{\eta} = \tilde{\eta}_0 h^3 + O(h^4) \]
\[ = -\frac{h^3}{15} \mathbb{E}(x_1^4 x_2) + O(h^4) \]
(12.116)

12.D Results of Table 12.2 and Table 12.3

Consider the first element in Table 12.2, i.e. \( R_d(1,1) \) in the EB approximation. Using Eqs. (12.74) and (12.107), it follows that
\[ R(1,1) = \mathbb{E}\{\tilde{x}_2^2\} \]
\[ = \mathbb{E}\{x_2 - \frac{h}{2} D x_2 + \frac{h^2}{6} D^2 x_2 + O(h^3)\}^2 \]
\[ = \mathbb{E}\{x_2\} - \frac{h}{2} \mathbb{E}(x_2 D x_2) + \frac{h^2}{3} \mathbb{E}(x_2 D^2 x_2) + \frac{h^2}{4} \mathbb{E}(D x_2)^2 + O(h^3). \]
(12.117)

Taking into account that \( \mathbb{E}(x_2 D x_2) = 0 \), cf. Remark 12.7, it follows that
\[ \tilde{R}_d(1,1) = \frac{h^2}{3} \mathbb{E}(x_2 D^2 x_2) + \frac{h^2}{4} \mathbb{E}(D x_2)^2 + O(h^3). \]
(12.118)

Similarly, the remaining elements of Table 12.2 can be systematically evaluated.

In Table 12.3, \( \tilde{R}_d(1,1) \) can be also expressed as a function of the system parameters exploiting the fact that
\[ D x_2 = \alpha x_1 + \beta x_2 + \gamma x_1^2 x_2. \]
(12.119)

Hence
\[ D^2 x_2 = D(\alpha x_1 + \beta x_2 + \gamma x_1^2 x_2) \]
\[ = \alpha x_2 + \beta D x_2 + \gamma(2x_1 x_2^2 + x_1 x_2^2). \]
(12.120)
Now reusing $Dx_2 = \alpha x_1 + \beta x_2 + \gamma x_1^2 x_2$ in (12.120) and substituting in (12.118), it follows that

$$\tilde{R}_d(1, 1) = h^2 \left[ \left( \frac{\alpha}{3} + \frac{7\beta^2}{12} \right) E(x_2^2) + \frac{\alpha^2}{4} E(x_1^2) + \frac{7\beta\gamma}{6} E(x_1^2 x_2^2) \right. \right.
\left. \left. + \frac{27}{3} E(x_1 x_2^3) + \frac{7\gamma^2}{12} E(x_1^4 x_2^3) \right]. \right.$$  

(12.121)

Following the same procedure, the remaining terms of Table 12.3 can be evaluated.

### 12.5 Asymptotic Discretization Bias for $S2$ Using $A3$

From Table 12.2

$$\tilde{r}_d - \tilde{R}_d \theta_0 = h^2 \left( \begin{array}{c} \frac{1}{4} E(x_2^3) + \frac{\alpha^2}{3} E(x_2^2) - \frac{\alpha}{3} E(x_1^2 x_2^2) \\ \frac{\alpha}{4} E(x_1^3) + \frac{\alpha^2}{3} E(x_2^2) - \frac{\alpha}{3} E(x_1^2 x_2^2) \\ \frac{\alpha}{4} E(x_1 x_2^3) + \frac{\alpha^2}{3} E(x_2^2) - \frac{\alpha}{3} E(x_1^2 x_2^2) \end{array} \right) + O(h^4),$$

(12.122)

$$R = \left( \begin{array}{ccc} E(x_2^2) + \frac{\alpha^2}{3} E(x_2^2) & \frac{\alpha^2}{3} E(x_1^2 x_2^2) + \frac{\alpha^2}{3} E(x_1^2 x_2^2) \\ \frac{\alpha^2}{4} E(x_2^2) & \frac{\alpha^2}{3} E(x_2^2) + \frac{\alpha^2}{3} E(x_1^2 x_2^2) \\ \frac{\alpha^2}{4} E(x_1 x_2^2) & \frac{\alpha^2}{3} E(x_1 x_2^2) + \frac{\alpha^2}{3} E(x_1 x_2^2) \end{array} \right) + O(h^4).$$

(12.123)

Inverting $R$ and substituting in (12.16) gives (neglecting $h^4$ terms)

$$\beta_d \approx \frac{h^2 E(x_1^2)}{3 |R|} \left[ \left( E(x_1 x_2^2) E(x_2^2) - E(x_1 x_2^2) E(x_2^2) \right) \right. \right.$$
\left. \left. + \frac{\beta}{2} \left( E(x_1 x_2^2) E(x_2^2) - E(x_1 x_2^2) E(x_2^2) \right) \right. \right.$$
\left. \left. + \frac{\gamma}{2} \left( E(x_1 x_2^2) E(x_2^2) - E(x_1 x_2^2) E(x_2^2) \right) \right], \right.$$

(12.124)

$$\bar{\alpha}_d \approx \frac{h^2}{3 |R|} \left[ \left( E(x_2^2) E(x_1 x_2^2) - [E(x_1 x_2^2)]^2 \right) \times \right. \right.$$
\left. \left. \left( E(x_1 x_2^2) - \frac{\beta}{2} E(x_1 x_2^2) - \frac{\gamma}{2} E(x_1 x_2^2) \right) \right], \right.$$

(12.125)
12.6. Asymptotic Discretization Bias for $S_2$ Using $A_3$

\[ \tilde{\gamma}_d \approx \frac{h^2 E(x_2^2)}{3|\mathbf{R}|} \left[ \left( E(x_2^2)E(x_1^2 x_2 D^3 x_2) - E(x_1^2 x_2^2)E(x_2 D^3 x_2) \right) \\
+ \frac{\beta}{2} \left( E(x_1^2 x_2^2)E(x_2 D^2 x_2) - E(x_2^2)E(x_1^2 x_2 D^2 x_2) \right) \\
+ \frac{\gamma}{2} \left( E(x_1^2 x_2^2)E(x_1^2 x_2 D^2 x_2) - E(x_2^2)E(x_1^2 x_2^2) \right) \right]. \tag{12.126} \]

where

\[ |\mathbf{R}| = E(x_1^2) \left( E(x_2^2)E(x_1^2 x_2^2) - |E(x_1^2 x_2^2)|^2 \right) + O(h^2). \tag{12.127} \]

For a small sampling period, $|\mathbf{R}|$ can be approximated again to be

\[ |\mathbf{R}| \approx E(x_1^2) \left( E(x_2^2)E(x_1^2 x_2^2) - |E(x_1^2 x_2^2)|^2 \right) = |\mathbf{R}_0|. \tag{12.128} \]

Substituting by (12.128) in Eqs. (12.124)-(12.126), results in

\[ \tilde{\beta}_d \approx \frac{h^2 E(x_1^2)}{3|\mathbf{R}_0|} \left[ \left( E(x_1^2 x_2^2)E(x_2 D^3 x_2) - E(x_1^2 x_2^2)E(x_1^2 D^3 x_2) \right) \\
+ \frac{\beta}{2} \left( E(x_1^2 x_2^2)E(x_1^2 x_2 D^2 x_2) - E(x_1^2 x_2^2)E(x_2 D^2 x_2) \right) \\
+ \frac{\gamma}{2} \left( E(x_1^2 x_2^2)E(x_1^2 x_2 D^2 x_2) - E(x_1^2 x_2^2)E(x_1^2 D^2 x_2) \right) \right]. \tag{12.129} \]

\[ \tilde{\alpha}_d \approx \frac{h^2}{3E(x_1^2)} \left( E(x_1 D^3 x_2) - \frac{\beta}{2} E(x_1^2 D^2 x_2) - \frac{\gamma}{2} E(x_1^2 D^2 x_2) \right), \tag{12.130} \]

\[ \tilde{\gamma}_d \approx \frac{h^2 E(x_2^2)}{3|\mathbf{R}_0|} \left[ \left( E(x_2^2)E(x_1^2 x_2 D^3 x_2) - E(x_1^2 x_2^2)E(x_2 D^3 x_2) \right) \\
+ \frac{\beta}{2} \left( E(x_1^2 x_2^2)E(x_2 D^2 x_2) - E(x_2^2)E(x_1^2 x_2 D^2 x_2) \right) \\
+ \frac{\gamma}{2} \left( E(x_1^2 x_2^2)E(x_1^2 x_2 D^2 x_2) - E(x_2^2)E(x_1^2 x_2^2) \right) \right]. \tag{12.131} \]

Hence, Eq. (12.76) is concluded from Eqs. (12.129)-(12.131).
Part III

User Aspects and Future Research
Chapter 13

User Aspects

13.1 Introduction

Two nonlinear approaches for modeling periodic signals were suggested in Part I and Part II of this thesis. Also, a review of the existing approaches to the same problem was given in Chapter 1. In this chapter, different user aspects for the two suggested approaches are discussed. In the next chapter some comments on the different approaches to the problem of modeling periodic signals are given, and some topics for future research are presented.

13.2 User Aspects for the Approach of Part I

In this section, different user aspects for the approach of modeling periodic signals based on the Wiener model structure are presented. The approach was studied in Chapters 2-4.

13.2.1 Modeled Signal

The approach of Part I can deal with a wide class of periodic signals. It is suitable for the cases where the modeled signal is affected particularly by underlying static nonlinear distortion. In this case, the approach can provide estimates for the fundamental frequency and the static nonlinearity. Also, the approach can track frequency and amplitude variations in the periodic signal, see Examples 2.3, 3.4, 4.3-4.4.

13.2.2 Choice of the Algorithm

Three algorithms were introduced in Part I, namely: Algorithm (2.9), Algorithm (3.9) and Algorithm (4.8). Algorithm (2.9) was introduced in [121] for joint estimation of the fundamental frequency of the periodic signal and the parameters of the output nonlinear function, cf. Chapter 2. In order to obtain a treatment of the local convergence properties of the RPEM of Chapter 2, the static gain of the Wiener model structure was fixed in a subinterval $I_n$ in the static block instead of being fixed in the linear block. This modification
resulted in Algorithm (3.9), cf. Chapter 3. The adaptive grid algorithm (Algorithm (4.8)) was introduced in Chapter 4 to recursively estimate the grid points in addition to the fundamental frequency and the parameters of the static nonlinearity. It was shown that Algorithm (4.8) gives more accurate estimates than Algorithm (3.9), cf. Fig. 4.4. This is due to the fact that Algorithm (4.8) has more freedom to select the grid points and hence a reduction of modeling errors is achieved.

13.2.3 Grid Points

The choice of the grid points for the approach of Part I is essential to obtain a good model accuracy, especially for Algorithm (2.9) and Algorithm (3.9). These grid points are chosen by the user depending on any prior information about the modeled signal. For example, the peak to peak value of the waveform helps to determine the grids range for each interval. Intervals with distorted amplitude such as the case of the flute data in Example 3.5, see Fig. 3.4(a) and Eq. (3.51), require a higher number of grid points to identify all the details of the static nonlinearity.

Estimating the grid points recursively in addition to the driving frequency and the parameters of the nonlinear output function was considered in Algorithm (4.8) resulting in an automatic grid point adaptation. One advantage is that less effort need to be spent on the selection of the grid points. Note that an initial value of the grid points need to be provided though. The improvement in the accuracy obtained in Example 4.2 by using Algorithm (4.8) instead of Algorithm (3.9), see Fig 4.4, indicates the importance of a good selection for the grid points.

Also, the performance of the algorithms was studied in Chapters 3-4 with the number of grid points, see Figures 3.6, 4.7. It can be concluded that the performance improves as the number of grid points increases to some extent. Beyond this extent, further increase of the grid points does not improve the performance. On the contrary, an unnecessary increase of the grid points may lead to a reduction of the accuracy due to over-parameterization.

13.2.4 Nonlinear Modeler

One of the main objectives of the approach of Part I is to estimate the static nonlinearity. Since the static nonlinearity is unknown, a nonlinear modeler is needed. In Section 1.4.2 different possibilities to parameterize the static nonlinearity were suggested.

In this thesis piecewise linear and piecewise quadratic polynomials were considered. It is expected that a piecewise polynomial is sufficient to obtain accurate models in case an enough number of grid points are considered. In the case of using a small number of grid points, it is more proper to consider higher order polynomials to reduce modeling errors. Needless to say considering other possibilities to parameterize the static nonlinearity, see Section 1.4.2, requires modification of the gradient calculations of the algorithms of Part I.
13.2.5 Additive Noise

The approach of Part I can handle - with good efficiency - periodic signals with low signal to noise ratio (SNR). This can be noticed from the results of Fig. 1.16(d). Also, note that this noise can be colored, see [121]. Further, since an output error approach is used, the sensitivity to high frequency modeling errors can be expected to be better than, e.g., LS methods, see [114].

13.2.6 Local Minima

Due to the complexity of the prediction error criterion (2.5), it is minimized by a Gauss-Newton method in this thesis. Hence, starting with good initial estimates is essential for the approach to guarantee convergence of the Gauss-Newton method to a local minimum of the criterion function. This was taken into consideration for example when the flute data were modeled in Example 3.5. Therefore, a good initial value of the fundamental frequency was obtained using the periodogram. Note that convergence to false local minima is a general problem when output error techniques are used.

13.2.7 Design Variables

The algorithms introduced in Part I require initialization of $P(t)$, $R_1(t)$, $r_2(t)$ and the forgetting factor $\lambda(t)$. The guidelines for choosing these quantities are discussed in detail in Chapter 5 of [69]. In general, the choice of these initial values follows the lines:

- $P(0)$ is usually chosen as $P(0) = \rho I$. A large value of $\rho$ makes the initial value of the parameter vector $\hat{\theta}(0)$ only marginally important. The parameter estimates in this case will change quickly in the transient period. A small value of $\rho$ means that much confidence is put in the prior knowledge of $\hat{\theta}(0)$. In this case $\hat{\theta}(t)$ will converge slowly.

- $R_1(t)$ should be chosen constant and diagonal, i.e. $R_1(t) = r_1 I$ if no detailed knowledge about the system dynamics is at hand. Note also that only the ratio $r_1/r_2$ will influence the parameter estimates. In case there is a need to speed up the convergence or the tracking of some parameters, the diagonal weights of the matrix $R_1(t)$ corresponding to these parameters are given higher values, cf. Examples 2.3, 3.4, 4.3-4.4.

- $\lambda(t)$ is very often chosen to grow exponentially using the relation

$$\lambda(t) = \lambda_o \lambda(t-1) + 1 - \lambda_o.$$  

This choice was suggested to achieve optimal asymptotic accuracy for reasonably long data sequences, see [69]. The values $\lambda(0) = 0.95$ and $\lambda_o = 0.99$ have been proved to be useful in many cases.
13.2.8 Computational Complexity

In general, the complexity of the algorithms (2.9), (3.9) and (4.8) of Part I is low. More specifically, it is monotonically increasing from Algorithm (2.9) to Algorithm (4.8). This follows as a consequence of increasing the size of the parameter vector. Also, the computational complexity of each individual algorithm increases as the number of grid points chosen by the user is increased.

13.3 User Aspects for the Approach of Part II

Different user aspects for the approach of modeling periodic signals using orbits of second-order nonlinear ODE’s are presented in this section. The approach was studied in detail in Chapters 5-12.

13.3.1 Modeled Signal

The approach of modeling periodic signals using a second-order nonlinear ODE model is suitable for periodic signals that have phase plots that lack any self intersections. As was discussed in Chapter 6, intersected periodic orbits in \( \mathbb{R}^2 \) needs higher order ODE’s to be modeled accurately. Also, the approach of Part II is (so far) not suitable for signals that are almost periodic, i.e. that contain amplitude or frequency variations. This is so since these signals do not represent stable periodic orbits. In this case the suggested nonlinear ODE model needs to be modified to track amplitude and frequency variations.

13.3.2 Choice of the Algorithm

Different off-line algorithms (the LS, the Markov, the ML and the LS-ASA) and on-line algorithms (the KF and the EKF) were introduced in Part II. The choice of the algorithm depends on some factors such as the SNR, polynomial degrees and the computational complexity. See Sections 13.3.4, 13.3.6, 13.3.7 for details.

In general, the LS-ASA algorithm (cf. Chapter 10) has the best performance as compared to other off-line algorithms. It also does not require initialization. The EKF algorithm can be recommended for on-line applications. Here it may be necessary to initialize the algorithm with parameters from an initial run of, e.g., the KF algorithm. The LS-ASA and the EKF algorithms can be used to model highly noise contaminated periodic signals. On the other hand, the rest of the algorithms introduced in Part II are more suitable for high SNR cases.

If possible, the Liénard model (cf. Chapter 11) is preferred as compared to the general signal model of Chapter 5. Apart from the reduction of the number of parameters, it allows for checking the existence of the periodic orbit generated by the model.
13.3. User Aspects for the Approach of Part II

13.3.3 Sampling Period

The sampling period is a critical parameter for the approach of Part II. As was indicated in Chapter 12, the total estimation bias in the LS algorithm is a combination of random noise errors \( \tilde{\theta}_n \) and discretization errors \( \tilde{\theta}_d \). Also, it is proved in Chapter 12 that

\[
\tilde{\theta}_n \propto \frac{1}{h^2} \quad \text{and} \quad \tilde{\theta}_d \propto h^2.
\]

Hence, there is an optimal sampling period that achieves the lowest estimation errors. Therefore, the sampling period must be chosen carefully. This can be done for example by plotting the mean square error of the modeling errors, i.e. the difference between the true signal and the model output, for different selections of the sampling period. A suitable sampling period can then be chosen as the value that achieves lowest mean square errors.

13.3.4 Additive Noise

The construction of the LS, the Markov and the KF algorithms introduced in Part II is based on the true states \( x_1(t) \), \( x_2(t) \) and \( \dot{x}_2(t) \). Since these states are unknown, they are replaced by their estimates. The estimates \( \hat{x}_1(t) \), \( \hat{x}_2(t) \) and \( \hat{\dot{x}}_2(t) \) were evaluated using finite difference approximations in Chapters 7-8. Therefore, due to the differentiation of the noise, these algorithms were biased for low and moderate SNRs. The situation was much better when the EKF algorithm was used in Chapter 8 since the states \( x_1(t) \) and \( x_2(t) \) were estimated recursively in addition to the parameters of the ODE model, cf. Fig. 8.3.

The automated spectral analysis (ASA) technique of Chapter 10 was introduced to extend the operating region of the off-line algorithms of Part II towards low SNRs. This was done by differentiating the signal - after eliminating the noise - in the frequency domain. This leads to much better results, cf. Fig. 10.1. Therefore, the user is advised to use finite difference approximations only for high SNR scenarios. In case of low SNR, the EKF or algorithms based on the ASA technique such as the LS-ASA of Chapter 10 are good candidates.

13.3.5 Design Variables

The design variables \( P(t) \), \( R_1(t) \) and \( r_2(t) \) are also needed for the implementation of the KF and the EKF algorithms. See Section 13.2.7 for details.

13.3.6 Polynomial Degrees

The approach of Part II is based on modeling the right hand side function of the second-order nonlinear ODE model using basis functions. In this thesis, the use of a polynomial basis was considered. The selection of the polynomial degrees \( L \) and \( M \), cf. Eqs. (5.25)-(5.26), is one of the major choices. Undermodeling and overparameterization lead to modeling errors and/or numerical problems.
This issue was addressed in Examples 7.3-7.4, 7.6, 8.3, see Figures 7.5-7.6, 7.10, 8.6.

Using too many parameters (overparameterization) normally means that the obtained accuracy is reduced. However, the effect of overparameterization becomes more severe in cases where the underlying optimization problem becomes singular because of an excessive number of parameters. In such situations algorithms such as the EKF algorithm may fail completely, which is a reason why a detailed understanding of overparameterization is central. Such a detailed analysis is a topic for future research, cf. Section 14.2.

In practical situations, the main problem is most often undermodeling, i.e. the imposed model is not capable of describing the system (signal) completely. For example, Fig. 6.2 was generated from a Liénard system with $L = 14$ and $M = 1$. Since the size of the parameter vector $\theta$ is a function in $L$ and $M$, it is not appropriate to use high polynomial degrees especially at low SNR. For this reason, the ODE model based on Liénard’s equation was suggested in Chapter 11.

In this thesis, two methods for model order selection were capable of handling undermodeling effects. The most straightforward method to select model order is perhaps to exploit plots of the measured signal together with simulated signals obtained by the estimated model. In this particular setting it is particularly suitable to use phase plane plots or periodograms to assess frequency domain properties, see Figures 8.2-8.3, 10.2-10.7.

The second useful method used in this thesis is to calculate the sum of the squared prediction errors for different selections of the model order. When this performance measure is plotted against the model order a suitable model order can often be selected where the performance measure levels out, or where it starts to increase, see Figures 7.5, 7.10.

Still another method could be to apply the mean residual analysis method of [120] for model order selection. That method assesses the amplitude error of the output signal.

The selection of the polynomial orders is a critical point in the approach of Part II. Hence, some of the topics for future research are to find a suitable model reduction technique or making use of other kinds of basis functions, see Section 14.2 for further details.

### 13.3.7 Computational Complexity

Most of the algorithms suggested in Part II such as the LS, the KF and the EKF algorithms have low computational complexities. On the other hand, the Markov and the ML algorithms have a quite high computational complexity. Due to the fact that the Markov estimation algorithm needs a matrix inversion and solution of a Lyapunov equation, see Eqs. (7.22), (7.38) and Example 7.5, it was difficult to extend the data length to 2000 samples or more to increase the accuracy of the estimated model.
On the other hand, the implementation of the ML estimation algorithm suffered from a lot of numerical difficulties. These difficulties prevented an increase of the number of Monte-Carlo simulations over 50 experiments in Examples 9.1-9.2. In order to obtain good accuracies for the estimated parameters (as compared to the CRB), see Examples 9.1-9.2 and Figures 9.1-9.2, the user is advised to follow the following recipe:

1. The sampling period should be as small as possible since this will reduce the errors in the ODE solver. Hence, a large data length is necessary to have a clear periodic oscillation.

2. There are some options that can be imposed on the Matlab ODE solvers to control the error tolerance such as “RelTol”, see e.g. ode45 help in Matlab.

3. The accuracy of the estimated signal is much better if “two” ODE solvers are considered. One for the estimated model and the second for Eqs. (9.33)-(9.40) which are used to calculate the sensitivity derivatives necessary to evaluate the gradient and the Hessian, cf. Eq. (9.21). This is because the accuracy of the ODE solver decreases when the dimension of the system of ODE’s is increased. Degradation in the accuracy of the ODE solver could lead to an oscillatory gradient instead of a decreasing one.

4. The parameter vector should contain only the effective parameters. Hence, 5 parameters only were considered in the simulation study of Chapter 9 instead of 11 parameters. This gives an indication of the importance to apply some model reduction techniques on the approach of Part II.

5. In order to study the performance of the ML algorithm versus the available data length, it is appropriate to use a complete number of periods. This is to provide estimates that are balanced with respect to the state space signal energy available for estimation. Otherwise, the ML method can be expected to be biased for finite data sets.

6. The gradient algorithm (9.21) needs a high number of iterations to converge as the data length increases.
Chapter 14

Conclusions and Future Research

14.1 Conclusions

The selection of a suitable method for modeling periodic signals is an application-dependent choice. Some applications require a complete or partial knowledge about amplitudes, frequencies and phases of the harmonic components. Some other applications require estimation of underlying nonlinearities or a mathematical model for the periodic signal.

Different approaches to periodic signal modeling were introduced and discussed in the previous chapters of this thesis. As was mentioned, the periodogram (cf. Section 1.3.1) is a fast but sometimes a poor estimator due to, e.g., bias, high variance, leakage and low spectral resolution. The modified periodogram methods have a medium resolution but this improvement is obtained on the expense of an increased bias. The periodogram can be used, as is done in Example 3.5, as an initial step to give a good initial estimate for the fundamental frequency of the measured data.

Line-spectra methods (cf. Section 1.3.2) such as the ESPRIT and the MUSIC algorithms give estimates with high accuracy on the account of the computational burden. Also, it is crucial to estimate the number of the sinusoidal components in these methods. Line-spectra methods are typically used in applications where sine waves in noise need to be detected and estimated. Some examples can be found in array signal processing, sonar and radar. It is crucial that the noise is white when these methods are applied, cf. Example 1.4.

Adaptive comb filtering techniques (cf. Section 1.3.3) can be used also for periodic signal modeling. Even if it is considered the most common approach to be used in many applications, it lacks the ability to give information on any underlying nonlinearities.

The nonlinear approach based on the Wiener model structure has two additional properties compared to other approaches. First, it gives information on the underlying nonlinearity, in cases where the overtones are generated by nonlinear imperfections in the system. The estimation of the underlying nonlinearity is an important issue in some applications of signal processing. For example, high-speed wireless communication channels often need a nonlinear equalizer for acceptable performance, see [72]. Secondly in some cases it may be known that the modeled signal closely resembles some other signal than a
sine wave, e.g. a square wave or a sawtooth wave. In this case prior information about the waveform can be used to increase the efficiency of the algorithm and the suggested method may then be more efficient than others.

On the other hand, the nonlinear approach based on the Wiener model structure lacks the ability to directly estimate the amplitudes and phases of the harmonic components. Also, as was mentioned in Chapter 1, the prediction error criterion used in this approach often has several local minima. Thus, it is important to choose good initial conditions to guarantee that the algorithm will converge to the true parameter vector.

The second nonlinear approach for periodic signal modeling of this thesis was introduced in Part II. The approach is based on using orbits of second-order nonlinear ODE’s. The motivation of suggesting this approach is the existence of many periodic systems that can be described by second-order nonlinear ODE’s. Moreover, many of these systems are described by polynomial right hand sides. Hence, polynomial basis functions were used to approximate the right hand function of the suggested nonlinear ODE model. By using the approach of Part II, a good opportunity exists to obtain highly accurate models by estimating only a few parameters.

On the other hand, the nonlinear approach of Part II lacks the ability to directly estimate the amplitudes and phases of the harmonic components. The technique is most suitable for periodic signals that do not intersect themselves in the $R^2$ phase plane. Also, the phase of the generated periodic signal from the ODE model is highly dependent on the accuracy of the estimated parameters. This means that small drifts from the true parameters cause a phase shift between the true and the model output signals even when the model signal resembles the true signal very well, cf. Fig. 10.7.

14.2 Topics for Future Research

There are several possibilities for further research that can be pursued within the framework provided by this thesis. Some of the possible extensions are shortly described below:

- It would be interesting to perform a convergence analysis of the adaptive grid point algorithm introduced in Chapter 4. The analysis would follow the main lines of the analysis of Chapter 3.

- The use of finite difference approximations in the LS estimation algorithm of Chapter 7 leads to biased estimates. This is because the regressors and the regressed variable of the linear regression equation are contaminated with noise, and the problem becomes an error-in-variables (EIV) problem, see [107]. It would be interesting to study different EIV techniques, especially the total least squares (TLS) [108] method, to find a suitable approach to get ride of this bias when a finite difference approximation is used.
• It would be interesting to check the possibility of using model reduction techniques, see [78], to reduce the size of the parameter vector to be estimated for the nonlinear ODE model. This is expected to increase the accuracy of the estimated model, and reduce the risk of divergence in case of using, e.g., the EKF algorithm.

• It is also interesting to search for a statistical method to detect the existence of the periodic orbit from the estimated parameter vector. This method can be used instead of only plotting the phase portrait of the solution of the estimated ODE model and comparing it to the phase plot of the true data. Extensions on the results based on Liénard’s equation may provide useful ideas.

• The approach of Part II uses polynomial basis functions to parameterize the right hand side of the second-order nonlinear ODE model. An interesting topic for future research is to study the possibility to use other kinds of basis functions, such as orthogonal functions or sinusoids. The goal will be to check if such basis functions would be better than the polynomial basis regarding stability of algorithms and numerical accuracy of the estimated model.

• The practically very important issue of model order selection in the approach using second-order nonlinear ODE model seems to require further research on identifiability properties to find the conditions that may result in an overparameterized model.

• Another interesting problem is to derive a convergence analysis for the ODE approach of Part II. An idea to do this is to perform an averaging analysis, see [5], for the suggested ODE model. Averaging is expected to simplify the convergence analysis considerably by assuming that the parameter vector is changing much more slowly than the states of the ODE model.

• It is of interest to modify the ODE approach to track amplitude and frequency variations in the modeled periodic signal. This will allow to model quasi periodic signals.
Svensk Sammanfattning

Vår omvärld är full av system som periodiskt återvänder till samma initialtillstånd, efter att ha passerat samma sekvens av mellantillstånd varje period. Vardagslivet är mer eller mindre uppbyggt av sådana fenomen; skiftningar mellan natt och dag, tidvattnets ebb och flod, månens olika faser och årstidernas årliga cyklar.

En signal $f_p(t)$ kallas periodisk om det för alla tidpunkter $t$ finns ett positivt reellt tal $T_0$ så att

$$f_p(t) = f_p(t + T_0).$$

Det minsta värdet på $T_0$ så att relationen är uppfylld är periodtiden för signalen $f_p(t)$.

Många fysikaliska signaler kan approximativt betraktas som periodiska. Exempel på detta är tal, musikaliska vågformer, akustiska vågor genererade av helikoptrar och båtar, och utsignaler från möjligtvis olinjära system som exiteras av en sinusformad insignal. Vid patientövervakning är många signaler som spelas in approximativt periodiska, t.ex. hjärtrytm, kardiogram och pneumogram.

Inom repetitiva reglerteknik tillämpningar, t.ex. utförande av något repetitivt arbete m.h.a. robotmanipulatorer, kan periodiska signaler användas som referenssignal till regulatorn. Periodiska signaler kan även användas för att detektera och skatta vibrationer i maskiner inom maskinteknik. Inom signalbehandling uppstår problemet att från brusförvanskade mätningar extrahera signaler som innehåller (nästan) periodiska komponenter. Typiska tillämpningar finns inom radar, sonar och telekommunikation. Där är det viktigt att skatta parametrar för dessa periodiska komponenter.


Den röda tråden i avhandlingen handlar om att använda olinjära tekniker för att modellera periodiska signaler. De föreslagna metoderna utnyttjar användarens a priori-kunskaper om signalens vågform. Förutsatt att den antagna modellstrukturen är lämplig ger detta teknikerna i denna avhandling en fördel gentemot andra angreppssätt, som inte förutsätter sådan kunskap.

I Del I vilar den föreslagna tekniken på det faktum att en sinusvåg som passerar genom en statisk olinjär funktion genererar ett harmoniskt spektrum av övertoner. Den skattade signalen kan följaktligen parametreras som en känd periodisk funktion (med okänd frekvens) i kaskad med en okänd statisk olinjäritet. Den okända frekvensen och den statiska olinjäritets parametrar
(Idén bakom de tekniker som föreslås i första delen kan sammanfattas som följer. Låt den drivande insignalen \( \hat{u}(t, \omega) \) definieras som
\[
\hat{u}(t, \omega) = \Lambda(\omega t),
\]
där \( t \) är diskret tid och \( \omega \in [0, \pi] \) är den okända, normaliserade vinkel-frekvensen. Dela sedan upp en hel period av \( \Lambda(\omega t) \) i \( L \) disjunkta intervall, \( I_j, j = 1, \ldots, L \). Om vi låter \( f_j(\theta_j, \Lambda(\omega t)) \) ange den statiska olinjäritet som används i intervall \( I_j \), så kan modellens utsignal beskrivas som
\[
\hat{y}(t, \omega, \theta_n) = f_j(\theta_j, \Lambda(\omega t)), \quad \Lambda(\omega t) \in I_j, \quad j = 1, \ldots, L
\]
\[
\theta_n = \begin{pmatrix} \theta_1^T & \theta_2^T & \cdots & \theta_L^T \end{pmatrix}^T.
\]
Om en styckvis linjär modell för parametrisering av \( f_j(\theta_j, \Lambda(\omega t)) \) används, definieras ett antal nät-punkter och elementen i parametervektorerna \( \theta_j \) väljs som värdena av \( f_j(\theta_j, \hat{u}(t, \omega)) \) i dessa punkter.

En rekursiv Gauss-Newton-skattare erhålls sedan genom att minimera kostnadsfunktionen
\[
V(\omega, \theta_n) = \lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} E \left[ y(t) - \hat{y}(t, \omega, \theta_n) \right]^2,
\]
där \( y(t) \) är den uppmätta signalen som skall modelleras.

En fullständig behandling av de lokala konvergensegenskaperna hos den föreslagna RPEM-algoritmen ingår i Del I. Dessutom presenteras en adaptiv nät-punkttsalgoritm för att estimeras den okända frekvensen samt parametrarna i de statiska olinjäriteterna för ett antal adaptivt estimerade nätpunkter. Detta ger RPEM-metoden större frihet att välja punkter i nätet och reducerar därför modelleringsefelet.

I Del II används en andra ordningens icke-linjär ODE för att modellera den periodiska signalen som en självsvängning, "limit cycle oscillation". Idén bakom denna metod är baserad på följande punkter:

- **Modellstruktur:**
  Den periodiska signalen modelleras som en funktion av tillstånden av en andra ordningens olinjär ODE
  \[
  \hat{y}(t) = f(y(t), \dot{y}(t), \theta),
  \]
där \( \theta \) är en okänd parametervektor som skall bestämmas.

- **Parametrisering av modellen:**
  Funktionen \( f(y(t), \dot{y}(t), \theta) \) kan då parametreras genom att skrivas som en summa av trunkerade polynom.
• **Diskretisering av modellen av modellen:**
  För att kunna implementera de olika algoritmerna diskretiseras ODE-modellen i tiden.

• **Estimering av parametervektorn:**
  Olika statistiskt baserade metoder för att estimera parametervektorn $\theta$ utvecklas.

Bibliography


