

UNIVERSITY OF CALGARY

Algorithms for Non-Convex System Identification

by

Seyed Pouyan Jazayeri

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL AND COMPUTER
ENGINEERING

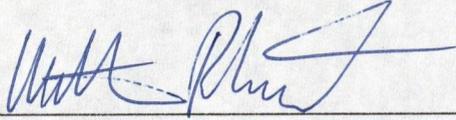
CALGARY, ALBERTA

APRIL, 2007

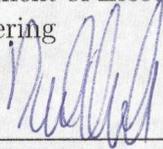
© Seyed Pouyan Jazayeri 2007

UNIVERSITY OF CALGARY
FACULTY OF GRADUATE STUDIES

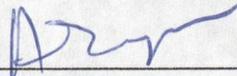
The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "Algorithms for Non-Convex System Identification" submitted by Seyed Pouyan Jazayeri in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE.



Supervisor, Dr. William Rosehart
Department of Electrical & Computer
Engineering



Co-supervisor, Dr. David Westwick
Department of Electrical & Computer
Engineering



Dr. Abraham Fapojuwo
Department of Electrical & Computer
Engineering



Dr. Hamidreza Zareipour
Department of Electrical & Computer
Engineering



Dr. Ian Gates
Department of Chemical & Petroleum
Engineering

April 9, 2007

Date

Abstract

This thesis investigates different approaches to non-convex system identification. Two algorithms are proposed in this document: a *multi-stage* identification algorithm and a *separable* identification algorithm. The multi-stage algorithm is a generalized approach which can be applied to any non-convex identification problem. The separable approach is a problem-specific technique requiring the identification problem to be convex in some of its parameters.

The problem of power system load modeling is investigated by using the proposed multi-stage and separable algorithms. Identification of human ankle dynamics is also explored by utilizing the proposed separable identification approach.

The proposed methodologies are verified by using artificial and real system data and are subsequently compared with published results. The numerical results presented demonstrate the effectiveness of the proposed algorithms.

Acknowledgements

I would like to acknowledge and thank my supervisors, Dr. William Rosehart and Dr. David Westwick. Their help and guidance, and above all, their abundance of patience and understanding, were instrumental in completion of my graduate studies.

I would also like to acknowledge all of my colleagues and the staff and professors at the Department of Electrical & Computer Engineering; They have been a constant source of inspiration and help. Dr. Antony Schellenberg, Dr. Codruta Roman and Mohammad Rasouli deserve special gratitude for donating their time, over the years, to help me in various aspects of my research

I am very fortunate to have had many sources of inspiration and support from outside of academia as well. Trevor, Matt, Richard, Len, Jennifer, and countless others have performed admirably in this prestigious role!

I am also very grateful to Ashley for years of patience, understanding and unconditional support.

Finally, I would like to acknowledge the financial contributions of the Alberta Research Council (ARC), Alberta Ingenuity Fund (AIF), and the Informatics Circle of Research Excellence (iCORE).

To my family;

This would not have been possible without their endless love and support.

Table of Contents

Approval Page	ii
Abstract	iii
Acknowledgements	iv
Table of Contents	vi
List of Tables	viii
List of Figures	ix
Nomenclature	xi
1 Introduction	1
1.1 Overview	1
1.2 Literature Survey	3
1.2.1 Power System Load Model Identification	4
1.2.2 Ankle Dynamics Identification	8
1.3 Research Motivation and Objectives	9
1.3.1 Power System Load Model Identification	9
1.3.2 Ankle Dynamics Identification	10
1.4 Implementation Methods	11
1.5 Thesis Structure	11
2 System Identification and Parameter Estimation	14
2.1 Introduction	14
2.2 Mathematical Formulation	17
2.3 Noise	18
2.4 Models for Non-linear Systems	20
2.4.1 Hammerstein Models	21
2.4.2 NARMAX Models	21
2.5 Identification Methods	24
2.5.1 Linear Regression and Ordinary Least Squares	24
2.5.2 Extended Least Squares	27
2.5.3 Levenberg-Marquardt Algorithm	32
2.6 Practical Applications	39
2.6.1 Power System Load Models	39

2.6.2	Parallel Pathway Model of Ankle Dynamics	43
2.7	Summary	45
3	Multi-stage System Identification	47
3.1	Introduction	47
3.2	Non-convex System Identification	47
3.3	Power System Load Model Identification	48
3.3.1	Proposed Multi-stage Identification Algorithm	48
3.3.2	Simulated System Results	56
3.3.3	Asymptotic Behaviour	59
3.3.4	Model Structure Verification	61
3.3.5	Field Data Results	71
3.4	Ankle Dynamics Model Identification	72
3.4.1	Proposed Multi-stage Identification Algorithm	73
3.4.2	Simulated System Results	82
3.5	Summary	85
4	System Identification with Variable Decomposition	86
4.1	Introduction	86
4.2	Convexity	88
4.2.1	Hessian	89
4.2.2	Exponential Expansion	91
4.3	Separation of Variables	96
4.4	Numerical Results	102
4.4.1	Performance Comparison	103
4.4.2	Simulated System Results	106
4.4.3	Sensitivity to Initial Values	108
4.4.4	Sensitivity to Theoretical Values	109
4.4.5	Application to Field Data	109
4.5	Summary	110
5	Conclusions and Summary	112
5.1	Overview	112
5.2	Contributions	113
5.3	Conclusions	114
	Bibliography	116

List of Tables

3.1	Parameter statistics for ASA and the Multi-stage approach from 30 Monte Carlo runs with $N = 1000$ points.	57
3.2	Simulated ankle parameters	82
3.3	Estimated parameters and residuals obtained from the Cross-Validation experiment	84
3.4	Ankle model parameters from 1000 Monte Carlo trials with SNR = 30 dB	85
4.1	Parameter Statistics and Time Performance of the RoC and Line Search Approaches for Finding θ_3	103
4.2	Parameter statistics for ASA and the separable approach from 30 Monte Carlo runs with $N = 1000$ points.	107
4.3	Estimated parameters and residuals obtained from the Cross-Validation experiment for the separable identification algorithm	108

List of Figures

2.1	A simple representation of a system with input u , disturbance w , and output y	15
2.2	The general system identification loop	16
2.3	A simple representation of a system with input u , noise v , and output y	19
2.4	The Hammerstein model for non-linear systems	21
2.5	Summary of the Extended Least Squares technique	33
2.6	The block structure version of the discrete-time dynamic load model	43
2.7	Continuous-Time model of ankle dynamics showing the relationship between the angle and the net torque of the ankle	44
3.1	The polynomial approximation to the discrete-time dynamic load model, used to derive the NARMAX representation.	50
3.2	Summary of the multi-stage identification process for the non-linear dynamic load model	56
3.3	The variance (var) and the Cramer-Rao Lower bound (CRLB) of the parameter estimates $(\theta_1, \theta_2, \theta_3)$ obtained from 5000 Monte Carlo runs of the proposed approach.	60
3.4	The 2-norm of residuals from approximating V^{θ_2} with 2 nd and 5 th order polynomials	62
3.5	The 2-norm of residuals when approximating V^{θ_2} with 2 nd and 5 th order polynomials (SNR = 30 dB)	63
3.6	Mean and variance of residuals when approximating V^{θ_2} with 2 nd and 5 th order polynomials (SNR = 30 dB)	64
3.7	Parameter distributions for the 2 nd order NARMAX model	69
3.8	Parameter distributions for the 5 th order NARMAX model	70
3.9	Error surface for field data with θ_2 fixed at 2.345. θ_1 and θ_3 are the varying parameters, and V_N is normalized 2-norm of the prediction errors	72
3.10	Discrete-Time model of ankle dynamics	81
3.11	Simulated and predicted output values for the dynamic ankle model	83
4.1	Eigenvalues, λ_1 and λ_2 , over the range of permissible θ_1 and θ_2 for $\theta_3 = 0.2$	92
4.2	Solution space over the range of permissible θ_1 and θ_2 for $\theta_3 = 0.2$	93
4.3	The first three terms in the regressor matrix after the exponential expansion of V^θ	95

4.4	Summary of the separable identification process for the non-linear dynamic load model	99
4.5	A hypothetical cost function over the range of θ_3 to demonstrate the pitfall of the RoC Method	100
4.6	Summary of the Improved Rate of Change identification process for the non-linear dynamic load model	101
4.7	Comparison of the values of θ_3 examined by the Line search and RoC approaches	105
4.8	Comparison of the values of θ_3 , near the optimal solution, examined by the RoC and Improved RoC approaches.	105
4.9	Measured and predicted active power, and prediction error, for the field data	111

Nomenclature

Acronyms

Name	Chapter(s)	Description
FIR	1	Finite Impulse Response
SLS	1	Separable Least Squares
ZIP	1	Constant impedance (Z), constant current (I), and constant power (P)
BFGS	1	Broyden, Fletcher, Goldfarb, and Shanno
ASA	1, 3, and 4	Adaptive Simulated Annealing
RC	1	Resistor-Capacitor
GA	1	Genetic Algorithms
ANFIS	1	Adaptive-Network-Based Fuzzy Inference System
EP	1	Evolutionary Programming
NARMAX	1,2, 3, and 5	Non-linear Auto Regressive Moving Average with exogenous inputs
iid	2 and 3	independent, identically distributed
LTI	2	Linear Time Invariant
ELS	2 and 3	Extended Least Squares
LM	2, 3, and 4	Levenberg-Marquardt
ZOH	2	Zero Order Hold
SNR	3 and 4	Signal-to-Noise Ratio
CRLB	3	Cramer-Rao Lower Bound
VAF	3 and 4	Variance Accounted For
RoC	4	Rate of Change

Power System Load Model

Name	Chapter(s)	Description
P_d	2	active power demand
P_0	2	nominal power
P (or y)	2, 3, and 4	normalized active power demand
z	2, 3, and 4	noise-contaminated active power demand
V_L	2	load voltage
V_0	2	nominal voltage
V	2, 3, and 4	normalized load voltage
x	2	internal state variable for P_d
T_p	2	time constant for P_d
N_{ps}	2	steady-state voltage index for P_d
N_{pt}	2	transient voltage index for P_d
θ	2, 3, and 4	load model parameters
θ_1	2, 3, and 4	steady-state voltage index
θ_2	2, 3, and 4	transient voltage index
θ_3	2, 3, and 4	inverse of load time constant
ℓ	2, 3, and 4	sampling period
Q_d	2	reactive power demand
z	2	internal state variable for Q_d
T_q	2	time constant for Q_d
N_{qs}	2	steady-state voltage index for Q_d
N_{qt}	2	transient voltage index for Q_d
ϕ	3	NARMAX parameters of the load model

Ankle Dynamics Model

Name	Chapter(s)	Description
$U(s)$	2 and 3	position (angle) of the ankle
$Y(s)$	2 and 3	net ankle torque
$V(s)$	2 and 3	ankle velocity
$W_L(s)$	2 and 3	intrinsic stiffness contribution to ankle torque
$W_{NL}(s)$	2 and 3	reflex stiffness contribution to ankle torque
I	2 and 3	inertia of the ankle joint
B	2 and 3	viscosity of the ankle joint
K	2 and 3	elasticity of the ankle joint
g	2 and 3	gain of reflex stiffness path
ζ	2 and 3	damping parameter of reflex stiffness path
ω	2 and 3	natural frequency of reflex stiffness path
Δ	2 and 3	delay of reflex stiffness path
T	3	discrete-time sampling period
τ	3	discrete-time delay of reflex stiffness path
ϕ	3	discrete-time NARMAX parameters

General

Name	Chapter(s)	Description
u	2	generic input
y	2	generic (uncontaminated) output
z	2	generic output
\hat{z}	2	predicted output
e	2,3, and 4	generic noise signal
n_u	2 and 3	maximum input lag in a NARMAX model
n_z	2 and 3	maximum output lag in a NARMAX model
n_e	2 and 3	maximum error lag in a NARMAX model
V_N	2, 3, and 4	cost function (or error/loss function)
N	2, 3, and 4	number of samples
ϵ	2, 3, and 4	residuals (prediction error)
ϕ	2, 3, and 4	(generic) system parameters
$\hat{\phi}$	2, 3, and 4	estimates of system parameters
ϕ^*	2, 3, and 4	optimal estimates of system parameters
J	2 and 3	Jacobian matrix
H	2, 3, and 4	Hessian Matrix
I	2	Identity matrix
q	2, 3, and 4	time shift operator
σ^2 or var	3 and 4	variance

Chapter 1

Introduction

1.1 Overview

The importance of accurate models in different fields of science and engineering can not be overstated. In fact, proper system identification is an essential component in a wide spectrum of applications such as biomedical systems, telecommunications, power systems, and control engineering. However, system identification can present a challenging task when pertaining to non-convex problems. If the system to be identified can not be represented as a linear regression, Finite Impulse Response (FIR) filter, or other simplistic models, then the identification problem becomes non-convex [39]. Since there are no “closed form” solutions for non-convex problems, they have been a source of continual research and exploration for years. In this thesis, the problem of non-convex system identification, in the context of power system load modeling and identification of human ankle dynamics, is explored.

The significance of accurate load models in power system analysis has been extensively documented [48]. The impact of load models on system stability including the system’s dominant eigenvalues, voltage stability, and inter-area oscillations has been extensively studied in the literature, for example [1, 5, 25, 53]. Also, as more power systems are operated with less stability margins, the importance of accurate load models increases. Because of the size and stochastic nature of loads, they are one of the most difficult elements in power systems to model. The challenge in power sys-

tem load identification is, in fact, twofold. First, a model that can accurately depict the load behaviour has to be chosen [44], then, an appropriate parameter estimation technique needs to be applied to the model of choice.

Creating mathematical representations for physiological systems from first principles and physical laws is often a difficult, if not impossible, task. As a result, system identification techniques are often the only tool for gaining insight into the structure and operation of physiological systems. An example of the application of system identification to physiological systems is the task of modeling human ankle dynamics. Accurate modeling of the dynamics of any joint, and specifically the ankle dynamics, can significantly improve the analysis of its posture and movement [60].

In this thesis, proposed identification algorithms are applied to a well-established power system load model and an ankle dynamics model. The two identification algorithms, a *multi-stage* approach and a *separable* identification technique, are proposed to estimate the parameters of non-convex systems. The multi-stage algorithm is a generalized approach that can be applied to any non-convex identification problem. The separable algorithm is a problem-specific technique which requires the identification problem to be convex in some parameters and non-convex in others.

The multi-stage and separable algorithms are applied to power system load model identification problem and the results are presented in this thesis. Also, the results from applying the multi-stage algorithm to the ankle dynamics identification problem are presented. To provide a foundation, background information on system identification is also included.

1.2 Literature Survey

Global optimization routines are sufficiently reliable, albeit very time consuming, tools for identifying non-convex systems. In other words, these routines will find the global optimum *if* given boundless time to operate. An example of this approach is found in [63] where the mechanical properties of lung tissue are examined. For this non-linear system, nonparametric time domain models and frequency domain models, and parametric block-structured models are studied. The global optimization routine is then utilized in order to identify the parametric block-structured models.

If global optimization is not acceptable, or practical, then a local optimization approach has to be utilized. Different variations of gradient descent algorithms are typically used for optimization, and the success of these routines depends on “good” initial values [39]. However, a number of alternative approaches to non-convex optimization also exist in the literature. For example, Separable Least Squares (SLS) optimization is used in [11] to identify Linear Time Invariant systems and the non-linear Wiener model. The authors show that the SLS approach to optimization can be numerically better conditioned – but not necessarily more efficient – than its Gauss-Newton counterpart. The authors also apply the proposed technique to identify the parameters of an industrial dryer and a high-purity distillation column. To obtain the initial estimates for the optimization routine, the authors use the subspace algorithms presented in [55].

For a few specific structures within the family of non-convex system identification problems, the optimal model can be *directly* obtained without resorting to global optimization techniques. For example, it is shown in [7] that an iterative identification

approach for Hammerstein models can converge to the global minimum. The authors demonstrate that the iterative technique – which oscillates between identifying the linear and non-linear parameters of the Hammerstein model – is generally convergent. In [64], the authors show that Hammerstein models with monotonic non-linearity can also be identified directly (and without the iterative approach of [7].) The authors demonstrate that if the system non-linearity is monotonic, then the identification task is reduced to a simple quadratic programming problem.

1.2.1 Power System Load Model Identification

Due to the complex nature of the load modeling problem, a variety of different solutions have been proposed in the literature. These approaches can be divided into two broad categories: *component-based* modeling, and *measurement-based* modeling. The component based approach, applies system identification to each individual load, whereas in the measurement-based approach identification is applied to signals from the aggregated system (i.e. at the substation level).

Examples of component-based techniques are presented in [29, 52]. In [52], an aggregated induction machine model is investigated. To derive the aggregate model, the parameters of all the induction motors connected to a single transmission line are required. The parameters of each individual machine are obtained by performing a number of tests on it. Once the individual machine data are collected and examined, a heuristic index value determines which machines are aggregated together in the final model. In [29], a different aggregated induction model is introduced for loads connected to a single substation. Parameters of individual motors and load composition data at each bus are required for this approach. These data are input

to an eigenvalue based classification scheme that determines the model values for the aggregated induction motor.

The measurement-based approach has also been utilized extensively on a number of different load models and parameter estimation techniques. Load models can be broadly categorized as any combination of non-linear/linearized and static/dynamic sets of equations. Estimation techniques can also be broadly categorized as system identification-based, exhaustive search, or neural networks-based methods. The remainder of this section presents a survey of the existing load model classes and estimation methods.

Static Models

Identification of static load models has received due attention in the academic literature. In [42], the most common static models, ZIP and exponential, are identified using a Weighted Least Squares approach. The authors acknowledge that the power system loads are dynamic (time-varying) in nature and, therefore, the model parameters have to be re-estimated for each new measurement. As a result, the weighting technique is utilized to reduce the impact of older measurements. In [18], the exponential static load model is identified by using a number of different gradient descent algorithms: the Newton Method, the BFGS (Broyden, Fletcher, Goldfarb, Shanno) Method, and modifications of each approach.

Dynamic Models

A non-linear dynamic load model is proposed in [25, 30] which aims to represent the aggregate effect of numerous load devices. The model contains non-linear differential equations and can represent steady-state and transient behavior of power system

loads.

Exhaustive Search Methods

In [34], the load model of [30] is linearized and then an exhaustive search algorithm, namely the Adaptive Simulated Annealing techniques, is utilized to determine the parameters of the linear model. Since the accuracy of the linearized model is shown to be less than ideal, the authors of [34] apply the Adaptive Simulated Annealing technique to find the parameters of the original non-linear dynamic load model. The results are consequently presented in [33]. Another application of exhaustive search to load modeling appears in [46]. In this paper, an iterative direct search process, called the Nelder Mead Simplex Search, is employed to find the parameters of a linearized (and frequency-dependent) dynamic load model. To obtain the measurements for this technique “Lock Tests” were performed on the system. In these tests, a small generator is tripped while a few other generators are locked at a given operating point (and hence unable to respond to the drop in frequency).

System Identification-based Methods

Examples of application of system identification techniques to load modeling can be found in [8, 14, 57, 61]. In [57], an RC circuit in parallel with an induction machine is used as the equivalent load model. The model is initially linearized, before a Newton-based gradient descent algorithm is used to determine its parameters. In [8], the non-linear dynamic load model of [30] is simplified to a polynomial linear regression model. Then, a Recursive Least Squares algorithm is employed to find the load parameters. This specific least squares algorithm was chosen so that the old measurement value are de-weighted and the most recent measurements are given more

significance. In [61], static load models along with a least squares curve fitting technique are used to predict the load behavior. Finally, in [14], a Quasi-Newton gradient descent algorithm, namely Levenberg-Marquardt, is used to identify the parameters of a number of different load models. In this paper, two different variations of the load model of [30], as well as a linear first order induction motor model, are studied.

Neural Networks-based Methods

Genetic Algorithms (GA) and Neural Networks-based approaches have also been used in identifying the parameters of static and dynamic load models. In [3], an Adaptive Neural Network technique is used to cluster a group of polynomial load models to arrive at an aggregated model, and the results are compared with conventional static load models. The clustering algorithm represented in the paper also involves a training process for the neural network. A Genetic Algorithms-based approach for power system load identification is presented in [28]. First, the authors introduce a composite static-dynamic load model based on previously established models. Then, a GA is applied to the proposed model to determine which parameters are significant before evaluating them. The algorithm consists of an arbitrary number of iterations – *generations* in GA terminology – of model evaluations for discovering the optimal set. In [58], the standard polynomial load model is identified by using a modified Genetic Algorithm routine. The authors use the stochastic search capability of a GA to examine a number of potential parameter sets before selecting a feasible answer. In [45], Fuzzy inference systems and Artificial Neural Networks techniques are combined together to produce the Adaptive-Network-Based Fuzzy Inference System (ANFIS) for load modeling. In this paper, no specific structure is assumed for the

load since the ANFIS, along with a hybrid learning algorithm, determines an appropriate polynomial model based on the measurement data. Finally, in [67], the load model of [30] is identified by using a GA-based algorithm. The authors use a randomly generated initial point and the major difference between the proposed approach and the standard GA technique is that the mutation component of the technique is based on Evolutionary Programming (EP).

1.2.2 Ankle Dynamics Identification

The current techniques for identifying the ankle dynamics model can be categorized into three different groups: Separable Least Squares approach, NARMAX approximation, and iterative identification technique. The NARMAX model is a general parametric representation for modelling nonlinear systems [37, 38].

In [32, 41], the authors propose an iterative approach for identifying the ankle model parameters. In these papers, the identification task is separated into two parts: the intrinsic dynamic component and the reflex pathway component. The proposed approach consists of iterative identification of each of these two parts until the problem converges. The inherent delays in the system make this approach to identification of the ankle dynamics possible. In other words, since the intrinsic component dies down before the reflex pathway has influenced the output, the identification of the two components can be separated.

In [59], a Separable Least Squares optimization is used to identify the ankle model. In this approach, the parameters of the model are divided into two groups: linear and non-linear. Then, it is shown that for a given non-linear set, identification of the linear parameters is reduced to a simple, and convex, linear regression. To

find the optimal non-linear parameters, a gradient descent optimization, namely the Levenberg-Marquardt, is used. The initial values for the Levenberg-Marquardt optimization are obtained from the approach presented in [32, 41].

In [36], the ankle dynamics are converted into a NARMAX model and are identified using an Extended Least Squares method. This transformation into the linear-in-the-parameters NARMAX model requires a few approximations within the system. The author shows that the NARMAX model can successfully model the output of the ankle dynamics model; however, the proposed approach finds a new set of parameters, namely the NARMAX parameters, which are not the same as those of the physical system.

1.3 Research Motivation and Objectives

The objective of this thesis is to develop algorithms that can be utilized in the identification of non-convex systems. The motivation and objectives are explored in detail in the context of the following problems: identification of power system load models and identification of human ankle dynamics.

1.3.1 Power System Load Model Identification

Emerging challenges with power system stability, and the general need for more accurate power system models, have emphasized the importance of accurate load models. The current methodologies in power system load model identification include a number of different approaches and techniques, as explored in the previous section. However, the existing methods are either very time consuming or limited in

some significant fashion. In fact, many utilities rely on worst case, lumped-bus load models such as the constant MVA [23]. The shortcomings of existing load modeling techniques are as follows:

- i) The component-based approach has the disadvantage of requiring information that is not generally available. In fact, the sheer volume of information required for this approach, as well as load ownership and confidentiality issues, make it very difficult, if not impossible, to implement the component-based approach.
- ii) Shortcomings of the Neural Network applications include slow convergence, difficulties in setting of the learning parameters, and training failure due to local minima. Genetic algorithms are stochastic in nature, which implies that they are not guaranteed to converge, even to a local minimum. Also, if high precision is needed, length of binary coded strings in GA will increase dramatically, thereby reducing the efficiency of the algorithm.
- iii) Finally, static load models fail to accurately represent power system behavior during and after contingencies.

This thesis proposes new identification algorithms for power system load identification that can overcome the aforementioned shortcomings in the current methodologies. More specifically, this thesis aims to develop new algorithms that can model power system loads both accurately and efficiently.

1.3.2 Ankle Dynamics Identification

The existing ankle dynamics identification techniques consist of approaches which require “good” initial values, [32, 41, 59], or do not identify the true parameters of

the ankle model, [36]. The objective of this thesis is to develop an algorithm that can identify the true parameters of the ankle model without dependence on initial values.

1.4 Implementation Methods

Unless otherwise stated, all procedures presented in this thesis are written by the author and implemented by using the MATLAB programming language and numerical computing environment. Existing MATLAB functions were utilized when solving linear and non-linear systems of equations and non-linear least squares curve-fitting. To solve a system of linear equations, which involves the computationally intensive task of matrix inversion, the `\` (left division) operator is used [19]. Since the `\` operator in MATLAB uses an LU factorization technique [20] which requires fewer floating-point operations and is significantly faster than a direct matrix inversion approach.

To solve a system of non-linear equations, the `fsolve` function in MATLAB is employed [19], and for non-linear least squares curve fitting problems, the `lsqcurvefit` function in MATLAB is utilized [15, 16].

1.5 Thesis Structure

The remainder of this thesis is structured as follows:

Chapter 2 provides an overview of some of the system identification and parameter estimation concepts that are fundamental to the algorithms

presented in this thesis. Background information is presented for the mathematical formulation of the system identification problem, along with the non-linear model structure used throughout the thesis (NARMAX). A few identification tools – least squares techniques and gradient descent optimization – are also discussed in detail. Finally, the load model and the ankle model explored in this thesis are presented.

Chapter 3 proposes a multi-stage identification algorithm and applies it to different identification problems. The proposed algorithm employs the following stages:

1. The non-convex identification problem is approximated with a linear-in-the-variables counterpart.
2. The optimal parameters for the new, convex problem are obtained.
3. These parameters are used as starting points in an optimization routine for the original, non-convex problem.

The algorithm is then applied to the problem of power system load model identification. It is shown to avoid the problems associated with local minima while being significantly more efficient than global search routines. Results are presented for artificial systems, as well as real (field) data. The multi-stage algorithm is also applied to the ankle dynamics identification problem. The algorithm is shown to

successfully determine the parameters of a simulated ankle joint.

Chapter 4 applies a separable identification algorithm to the power system load identification problem. This approach is a problem-specific technique and requires the problem to be convex in one or more parameters. Once the convexity of the power system load modeling problem is demonstrated in a subsection of its parameters, the proposed algorithm is applied to the problem. Once again, the proposed algorithm is shown to avoid the problems associated with local minima while being significantly more efficient than global search routines. Also, results are presented for artificial and real systems.

Chapter 5 summarizes the content of this thesis and provides conclusions. The significant components of the thesis are reviewed and the main contributions are highlighted.

Chapter 2

System Identification and Parameter Estimation

2.1 Introduction

In order to establish a foundation for the topics presented in this thesis, background information is provided in this chapter. More specifically, this chapter presents the non-linear models employed in this thesis, the identification methods used in the proposed algorithms, and the formulation of the system identification problem. Lastly, the power system load model and the ankle dynamics model which will be identified in this thesis are introduced.

In general, a *system* is an entity in which the interaction of different variables (system parameters) results in detectable signals. These signals can be divided into three categories: input, disturbance, and output. Inputs typically refer to signals that can be influenced by the observer, disturbances refer to inputs that can not be controlled by the observer, and outputs refer to signals of interest produced by the system in response to the input and disturbances [39]. A simplified graphical representation of a system is shown in Figure 2.1, where u is the input to the system, w is the disturbance, and y is the output.

A system *model* combines the observable signals from a system in some pattern. It indicates the relationship between the system's variables and its observable signals. Models range in a variety of classifications, including but not limited to: mental, graphical, and mathematical [39]. However, in this document, *model* will only refer

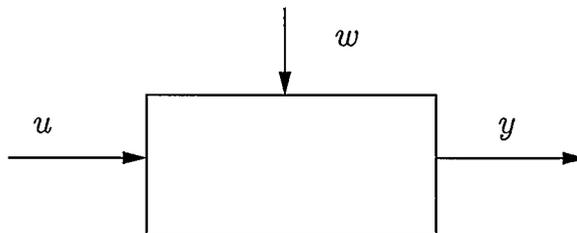


Figure 2.1: A simple representation of a system with input u , disturbance w , and output y

to mathematical (or analytical) models.

Mathematical models use mathematical expressions such as difference or differential equations to represent the relationship between the system's variables and its observable signals. These expressions can be further categorized to indicate the type of difference or differential equation used. Some common examples include linear and non-linear, or discrete-time and continuous-time [39]. For example,

$$y(t) + ay(t - 1) = b_0u(t) + b_1u(t - 1) \quad (2.1)$$

is a discrete-time, linear, difference equation model, where a , b_0 , and b_1 are the system parameters, and u and y are the system input and output, respectively. The mathematical representation of disturbances is intentionally left out of the model in (2.1), as it will be discussed later in Section 2.3.

An example of a continuous-time, differential equation model is:

$$\begin{aligned} \dot{x}(t) &= a_0x(t) + a_1u(t) \\ y(t) &= b_0x(t) + b_1u(t) \end{aligned} \quad (2.2)$$

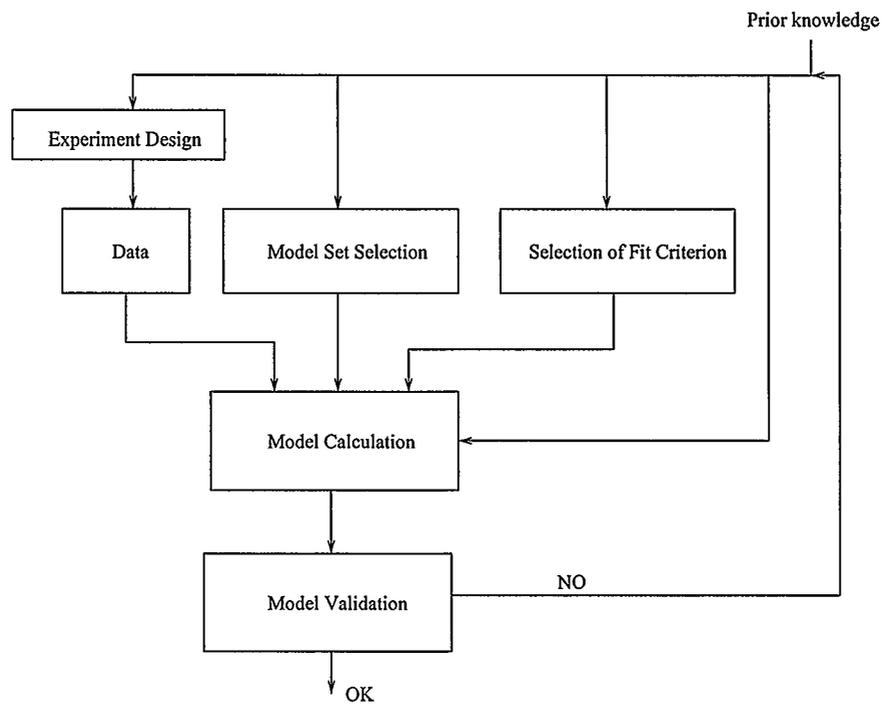


Figure 2.2: The general system identification loop

where $a_{0,1}$ and $b_{0,1}$ are the system parameters, x is an internal state variable and u and y are the system input and output, respectively.

System identification refers to the process in which the input and output measurements are recorded and analyzed in order to arrive at a system model. It should be noted that parameter estimation is a subset of the broader system identification problem. The typical system identification loop is shown in Figure 2.2 [39]. The following is a brief discussion of the components of this loop.

The main goal in the *Experiment Design* block is to arrive at the most suitable choices regarding the input and output signals. More specifically, the observer will investigate the choice of signals to be measured, their sampling rates, their times of measurement, and other similar aspects, in order to obtain the set of input/output

data that are most pertinent and beneficial to identifying the system. Clearly, all the decisions have to be made within the constraints of the problem at hand. Also, one can encounter scenarios in which the choice of data can not be manipulated by the observer and the only available data are obtained from the normal operation of the system.

During *Model set selection*, different model collections are investigated in order to determine the most suitable candidate set. This is often the most difficult part of the identification process. A combination of *a priori* knowledge, engineering intuition, and formal properties of models need to be combined in order to determine which set of models will be examined.

Selection of fit criterion determines how the candidate models will be assessed in order to obtain the most suitable model. Typically, a model's ability to reproduce measured data is the main criterion in evaluating the candidate models.

In *Model Calculation*, the selection of fit criterion along with the data set obtained from the Experiment Design phase, is applied to the candidate model set. Then, the signals predicted by the candidate models are calculated using an appropriate tool.

Finally, *Model Validation* compares the predicted and measured signals in order to determine the effectiveness of the candidate models [39].

2.2 Mathematical Formulation

In general, a model structure is a parametrized mapping from the measured data to the predicted outputs. In mathematical terms,

$$\hat{y}(\phi) = f\left(\phi, \{u(1) \cdots u(N), y(1) \cdots y(N)\}\right) \quad (2.3)$$

where ϕ is the model parameter set, u and y are the measured input and output, respectively, N is the number of measured samples of u and y , and \hat{y} is the predicted output. For example, if (2.1) is the true representation of a system, then

$$\hat{y}(t, \phi) = \phi_1 y(t-1) + \phi_2 u(t) + \phi_3 u(t-1) \quad (2.4)$$

is a candidate model to predict the output of the system. In the ideal case, the system identification process will find the candidate parameter set $\hat{\phi} = [\phi_1 \ \phi_2 \ \phi_3]$ to be equal to $[-a \ b_0 \ b_1]$. In other words, the aim is to find the candidate parameter set $\hat{\phi}$ such that :

$$\hat{\phi} = \arg \min_{\phi} V_N(\phi) \quad (2.5)$$

where $V_N(\phi)$ is a measure of the difference between the predicted output $\hat{y}(\phi)$ and the measured output y . One of the most common tools for solving the optimization problem of (2.5) is the Least Squares method, which will be presented in Section 2.5.

2.3 Noise

The term *noise* is traditionally used to designate unwanted signals that tend to disturb the operation of a system and over which the observer has little or no control. Measurement noise and process noise are the two most significant sources of disturbance (or contamination.) Examples of measurement noise include quantization

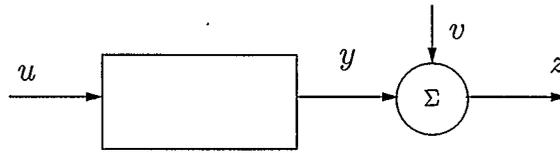


Figure 2.3: A simple representation of a system with input u , noise v , and output y . This diagram illustrates a system where an input u is processed by a block to produce an intermediate output y . This intermediate output y is then added to a noise signal v at a summing junction (represented by a circle with the Greek letter Σ) to produce the final measured output z .

errors in Analog-Digital conversion, interference from thermal noise, and intrusion of unwanted frequency components such as 60 Hz or AM radio [13, 24]. Examples of process noise include unmeasured inputs to the system such as turbulence when identifying the dynamics of an aircraft [39]. Also, modelling errors are often treated as noise. For example, the error introduced by linearizing a non-linear system can be treated as additive noise.

In general, noise is included in the system model as an output additive signal, as shown in Figure 2.3, where v can represent a variety of disturbances and/or modeling errors. System input, uncontaminated output, and measured output are represented by u , y , and z respectively.

Regardless of their origin, all types of noise have a common property: it is very difficult, if not impossible, to specify their magnitude as a function of time in precise terms. Therefore, common practice is to model the noise as an independent, identically distributed (*iid*) sequence with zero mean that has been filtered by a linear system. A sequence of independent, identically distributed random variables is also called *white noise* and is typically represented as $e(t)$. For example, if the system represented in (2.1) is contaminated with additive white noise, the input-output relationship becomes:

$$\begin{aligned}
z(t) &= y(t) + e(t) \\
&= az(t-1) + b_0u(t) + b_1u(t-1) + e(t)
\end{aligned} \tag{2.6}$$

and the equation for predicted output, (2.4), becomes:

$$\hat{z}(t, \phi) = \phi_1 z(t-1) + \phi_2 u(t) + \phi_3 u(t-1) \tag{2.7}$$

Once the output is contaminated, the true system output, $y(t)$, is no longer available. Therefore, the contaminated signal, $z(t)$, is used as the system output for identification purposes. Consequently, the generic model structure of (2.3) has to be modified to:

$$\hat{z}(\phi) = f\left(\phi, \{u(1) \cdots u(N), z(1) \cdots z(N)\}\right) \tag{2.8}$$

2.4 Models for Non-linear Systems

Non-linear systems can be described by a number of different models. Two common categories of models are: Wiener-Hammerstein models and parametric non-linear models, the most general of which is the NARMAX model. The Hammerstein model is acknowledged in the derivation of the discrete-time power system load model, and is introduced in Section 2.4.1. The NARMAX model will be used extensively throughout this document and is introduced in Section 2.4.2.

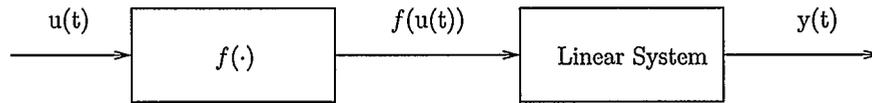


Figure 2.4: The Hammerstein model for non-linear systems

2.4.1 Hammerstein Models

Hammerstein models belong to the block-oriented group of models for non-linear systems, which consist of interconnections of Linear Time Invariant (LTI) systems and static (memoryless) non-linearities [39]. A system is time-invariant if a delay or advance of the input signal results in an identical shift on the output signal. In other words, the characteristics of such system do not change with time [24]. A system is memoryless if the output only depends on the present input (and not the past or future values).

A Hammerstein structure contains a static non-linearity $f(\cdot)$ at the input, followed by a linear system, as shown in Figure 2.4. A simple example for the static non-linearity of Figure 2.4, $f(\cdot)$, is a polynomial (of degree l):

$$f(u(t)) = a_0 + a_1u(t) + a_2u^2(t) + \dots + a_lu^l(t) \quad (2.9)$$

2.4.2 NARMAX Models

The Non-linear Auto Regressive Moving Average with eXogenous inputs (NARMAX) is a general parametric form for modeling non-linear systems. A NARMAX model uses a non-linear difference equation to describe the relationship between the system inputs and outputs [9], which can be represented mathematically as:

$$z(t) = f^l[z(t-1), \dots, z(t-n_z), u(t), \dots, u(t-n_u), e(t-1), \dots, e(t-n_e)] + e(t) \quad (2.10)$$

where f^l is a non-linear mapping, l is the non-linearity order, u , z , and e are the input, output and additive noise sequences, respectively. The maximum input lag, the maximum output lag, and the maximum error lag are shown as n_u , n_z , and n_e , respectively. In other words, n_u indicates how many past values of the input are required to correctly model the current output. In this document, f^l is only represented by a polynomial of degree l .

As an example, a polynomial NARMAX model with $n_u = n_z = n_e = 1$ (maximum lag on the signals is 1) and $l = 2$ (maximum polynomial degree is 2) is represented as:

$$\begin{aligned} z(t) = & a_0 + a_1z(t-1) + a_2u(t) + a_3u(t-1) + a_4e(t-1) + a_5z(t-1)u(t-1) + \\ & a_6z(t-1)e(t-1) + a_7u(t-1)e(t-1) + a_8z^2(t-1) + a_9u^2(t) + \\ & a_{10}u^2(t-1) + a_{11}e^2(t-1) + a_{12}u(t)u(t-1) + a_{13}u(t)e(t-1) + e(t) \end{aligned}$$

and in the vector format as:

$$z(t) = \begin{bmatrix} 1 \\ z(t-1) \\ u(t) \\ u(t-1) \\ e(t-1) \\ z(t-1)u(t-1) \\ z(t-1)e(t-1) \\ u(t-1)e(t-1) \\ z^2(t-1) \\ u^2(t) \\ u^2(t-1) \\ e^2(t-1) \\ u(t)u(t-1) \\ u(t)e(t-1) \end{bmatrix}^T \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \end{bmatrix} + e(t) \quad (2.11)$$

which contains 14 (potential) parameters to be identified. The terms in (2.11) are obtained as follows:

1. Since the maximum lag is 1 ($n_u = n_z = n_e = 1$), the following signals have to be represented in the NARMAX model:

$$u(t), \quad u(t-1), \quad y(t-1), \quad \text{and} \quad e(t-1) \quad (2.12)$$

2. Since the maximum polynomial degree is 2 ($l = 2$), all possible cross-product terms of 1st and 2nd order involving the signals in (2.12) have to be accounted

for in the NARMAX model.

It can be seen that the number of potential terms in a NARMAX model can rapidly increase when the order of the model increases. The maximum number of parameters in a NARMAX model, p , is calculated from [9] :

$$p = \sum_{i=0}^l p_i + 1 \quad (2.13)$$

where

$$\begin{aligned} p_0 &= 1 \\ p_i &= \frac{p_{i-1}(n_u + n_z + n_e + 1)}{i} \end{aligned}$$

For example, a model with $n_u = n_z = n_e = 4$ and $l = 2$ has $p = 106$, i.e. 106 parameters. It should be noted that NARMAX models based on real systems might have fewer parameters than the maximum number, p .

2.5 Identification Methods

2.5.1 Linear Regression and Ordinary Least Squares

Linear regression model structures are a convenient tool for representing the behaviour of basic linear and non-linear systems [39]. These structures refer to linear or non-linear parametric systems that can be described as:

$$z(t) = \mathbf{x}(t)^T \boldsymbol{\phi} + e(t) \quad (2.14)$$

where $z(t)$ is the output, $\mathbf{x}(t)$ is a *data dependent* vector containing system measurements, ϕ are the system parameters, and $e(t)$ is the additive noise. It should be noted that $\mathbf{x}(t)$ is also called the *regressor vector*.

For example, the system represented in (2.6) has a linear regression model structure for which the regressor vector, $\mathbf{x}(t)$, and the parameter vector, ϕ , are :

$$\mathbf{x}(t) = \begin{bmatrix} z(t-1) \\ u(t) \\ u(t-1) \end{bmatrix} \quad \text{and} \quad \phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} = \begin{bmatrix} a \\ b_0 \\ b_1 \end{bmatrix} \quad (2.15)$$

Since most system measurements are taken at more than one time point, the relationship in (2.14) can be extended to a sequence of measurements:

$$\mathbf{z} = \mathbf{X}\phi + \mathbf{e} \quad (2.16)$$

where \mathbf{z} is a vector of output measurements, \mathbf{X} is a matrix of system measurements (regressor matrix), and \mathbf{e} is a vector of additive noise. For example, if the input and output measurements of the system in (2.6) are recorded for the time interval $t = \{0 \dots N\}$, the regressor matrix becomes:

$$\mathbf{X} = \begin{bmatrix} z(-1) & u(0) & u(-1) \\ z(0) & u(1) & u(0) \\ z(1) & u(2) & u(1) \\ \vdots & \vdots & \vdots \\ z(N-2) & u(N-1) & u(N-2) \\ z(N-1) & u(N) & u(N-1) \end{bmatrix} \quad (2.17)$$

The goal of the system identification process is to find the parameter set, ϕ , that would minimize the prediction error – the difference between the measured output and that predicted by the model. For the generic system of (2.16), the system identification process will consist of finding a parameter set, $\hat{\phi}$, and the corresponding predicted output, \hat{z} , that are “as close as possible” to the real system parameters, ϕ , and the measured output, z . For the generic system of (2.16), the predicted output is:

$$\hat{z} = \mathbf{X}\hat{\phi} \quad (2.18)$$

where $\hat{\phi}$ is a set of predicted system parameters. Therefore, the *optimal* parameter set, ϕ^* should meet the following criterion:

$$\phi^* = \arg \min_{\phi} V_N(\phi) \quad (2.19)$$

where

$$V_N(\phi) = \frac{1}{2} \|z - \hat{z}(\phi)\|_2^2 \quad (2.20)$$

and $\|\boldsymbol{x}\|_2$ is the 2-norm (Euclidean norm) of \boldsymbol{x} .

Using the definition of Euclidean norm [2], (2.20) can be written as:

$$V_N(\boldsymbol{\phi}) = \frac{1}{2} \left((z - \hat{z}(\boldsymbol{\phi}))^T (z - \hat{z}(\boldsymbol{\phi})) \right) \quad (2.21)$$

Next, substituting the value for $\hat{z}(\boldsymbol{\phi})$ from (2.18) gives:

$$\begin{aligned} V_N(\boldsymbol{\phi}) &= \frac{1}{2} \left((z - \mathbf{X}\hat{\boldsymbol{\phi}})^T (z - \mathbf{X}\hat{\boldsymbol{\phi}}) \right) \\ &= \frac{1}{2} \left(z^T z - z^T \mathbf{X}\hat{\boldsymbol{\phi}} - (\mathbf{X}\hat{\boldsymbol{\phi}})^T z + (\mathbf{X}\hat{\boldsymbol{\phi}})^T (\mathbf{X}\hat{\boldsymbol{\phi}}) \right) \end{aligned} \quad (2.22)$$

Now, the problem of (2.19) can be regarded as an unconstrained optimization in the $\boldsymbol{\phi}$ parameter space. Therefore, the optimal point, $\boldsymbol{\phi}^*$, should meet the First Order Necessary Condition [10]: Setting the derivative of (2.22) (with respect to $\boldsymbol{\phi}$) to zero gives the relationship for the optimal parameter set, $\boldsymbol{\phi}^*$, as :

$$\mathbf{X}^T \mathbf{X} \boldsymbol{\phi}^* - \mathbf{X}^T z = 0 \quad (2.23)$$

$$\therefore \boldsymbol{\phi}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T z \quad (2.24)$$

This is the (Ordinary) Least Squares method for solving linear regression problems.

2.5.2 Extended Least Squares

If the regressor matrix \mathbf{X} contains lagged values of the disturbance term, $e(t)$, an Ordinary Least Squares approach can not be used for parameter estimation, as demon-

strated with the following example: in the NARMAX model of (2.11):

$$\begin{aligned}
 z &= \mathbf{X}\boldsymbol{\phi} + \mathbf{e} \\
 &= \begin{bmatrix} \vdots & \dots \\ e(t-1) & \dots \\ \vdots & \dots \\ z(t-1)e(t-1) & \dots \\ u(t-1)e(t-1) & \dots \\ \vdots & \dots \\ e^2(t-1) & \dots \\ \vdots & \dots \\ u(t)e(t-1) & \dots \end{bmatrix}^T \boldsymbol{\phi} + \mathbf{e} \quad (2.25)
 \end{aligned}$$

the regressor matrix contains 5 columns involving lagged values of $e(t)$. Since disturbance measurements, \mathbf{e} , are not available, they are replaced with prediction error, $\boldsymbol{\epsilon}$, where:

$$\boldsymbol{\epsilon}(t, \boldsymbol{\phi}) = z(t) - \hat{z}(t, \boldsymbol{\phi}) \quad (2.26)$$

Therefore, all the columns in \mathbf{X} that involve $e(t)$ are replaced with $\boldsymbol{\epsilon}(t, \boldsymbol{\phi})$. Consequently, the regressor matrix is a function of $\boldsymbol{\phi}$: \mathbf{X} is replaced with $\mathbf{X}(\boldsymbol{\phi})$, and the input-output relationship of (2.25) now becomes:

$$\begin{aligned}
z &= \mathbf{X}(\phi)\phi + e \\
&= \begin{bmatrix} \vdots & \dots \\ \epsilon(t-1, \phi) & \dots \\ \vdots & \dots \\ z(t-1)\epsilon(t-1, \phi) & \dots \\ u(t-1)\epsilon(t-1, \phi) & \dots \\ \vdots & \dots \\ \epsilon^2(t-1, \phi) & \dots \\ \vdots & \dots \\ u(t)\epsilon(t-1, \phi) & \dots \end{bmatrix}^T \phi + e \quad (2.27)
\end{aligned}$$

Based on (2.27), the predicted output is obtained from:

$$\hat{z} = \mathbf{X}(\hat{\phi})\hat{\phi} \quad (2.28)$$

The relationships of (2.27) and (2.28) look familiar to those of a linear regression problem, (2.16) and (2.18). However, due to the non-linear effect of ϕ on the regressor matrix, the parameter estimation problem is no longer a linear regression and can not be solved by using the ordinary least squares approach. In fact, the new problem is called a *pseudo-linear regression* and the Extended Least Squares (ELS) approach is used to solve it [9, 22].

A summary of the ELS algorithm is as follows:

1. A reduced regressor matrix $\tilde{\mathbf{X}}$ is formed by removing the columns of the regressor matrix $\mathbf{X}(\phi)$ that do not involve the error terms, $\epsilon(\phi)$. Next, the corresponding parameter vector, $\tilde{\phi}$, is defined by removing the parameters associated with the error terms from ϕ . For example, for the NARMAX model of (2.11) and (2.27), $\tilde{\mathbf{X}}$ and $\tilde{\phi}$ are:

$$\tilde{\mathbf{X}} = \begin{bmatrix} 1 & \cdots \\ z(t-1) & \cdots \\ u(t) & \cdots \\ u(t-1) & \cdots \\ z(t-1)u(t-1) & \cdots \\ z^2(t-1) & \cdots \\ u^2(t) & \cdots \\ u^2(t-1) & \cdots \\ u(t)u(t-1) & \cdots \end{bmatrix}^T \quad \tilde{\phi} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_5 \\ a_8 \\ a_9 \\ a_{10} \\ a_{12} \end{bmatrix} \quad (2.29)$$

Then the following linear regression problem

$$\tilde{\mathbf{X}}\tilde{\phi} = z$$

is solved using ordinary Least Squares to find the optimal values for $\tilde{\phi}$:

$$\tilde{\phi} = (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T z$$

2. Next, the *residuals*, ϵ , are calculated as:

$$\epsilon = z - \tilde{X}\tilde{\phi}$$

3. The iteration number, k , is set to 1.

4. $\mathbf{X}(\phi_{k-1})$ is formed by using $\epsilon(\phi_{k-1})$ as an estimate of the prediction error terms. For example, for the NARMAX model of (2.27), the regressor matrix for the k^{th} iteration of the ELS algorithm, $\mathbf{X}(\phi_{k-1})$, is given by:

$$\mathbf{X}(\phi_{k-1}) = \begin{bmatrix} \vdots & \cdots \\ \epsilon(t-1, \phi_{k-1}) & \cdots \\ \vdots & \cdots \\ z(t-1)\epsilon(t-1, \phi_{k-1}) & \cdots \\ u(t-1)\epsilon(t-1, \phi_{k-1}) & \cdots \\ \vdots & \cdots \\ \epsilon^2(t-1, \phi_{k-1}) & \cdots \\ \vdots & \cdots \\ u(t)\epsilon(t-1, \phi_{k-1}) & \cdots \end{bmatrix}^T \quad (2.30)$$

5. $\mathbf{X}(\phi_{k-1})\phi_k = z$ is solved using linear regression to determine the optimal values for ϕ_k , i.e.

$$\phi_k = \left(\mathbf{X}(\phi_{k-1})^T \mathbf{X}(\phi_{k-1}) \right)^{-1} \mathbf{X}(\phi_{k-1})^T z$$

6. New set of residuals is calculated as: $\epsilon(\phi_k) = z - \mathbf{X}(\phi_{k-1})\phi_k$
7. Set $k = k + 1$.
8. If the problem has converged, exit. Otherwise, go to Step 4. The convergence criterion is based on the change of ϕ between two sequential iterations, i.e. if $\|\phi_k - \phi_{k-1}\|_\infty$ is sufficiently small. (In this document, a value of 10^{-6} was chosen.) Even though the convergence of the ELS technique is not guaranteed, it has always converged to an optimal solution in the author's experiments as well as numerous examples in the literature [9, 50, 65, 66].

A flowchart of the ELS technique is shown in Figure 2.5.

2.5.3 Levenberg-Marquardt Algorithm

As previously shown in (2.20), the *cost function* in system identification is defined as:

$$\begin{aligned} V_N(\phi) &= \frac{1}{2} \|z - \hat{z}(\phi)\|_2^2 \\ &= \frac{1}{2} \|\epsilon(\phi)\|_2^2 \end{aligned} \tag{2.31}$$

If the predicted output relationship, $\hat{z}(\phi)$, can be represented in the linear regression format: $\hat{z} = \mathbf{X}\phi$, or pseudo-linear regression format: $\hat{z} = \mathbf{X}(\phi)\phi$, then the Ordinary Least Squares and Extended Least Squares techniques can be used, respectively, to minimize the loss function and obtain the optimal parameter set. The linear and pseudo-linear regression models are also known as (*pseudo*) *linear-*

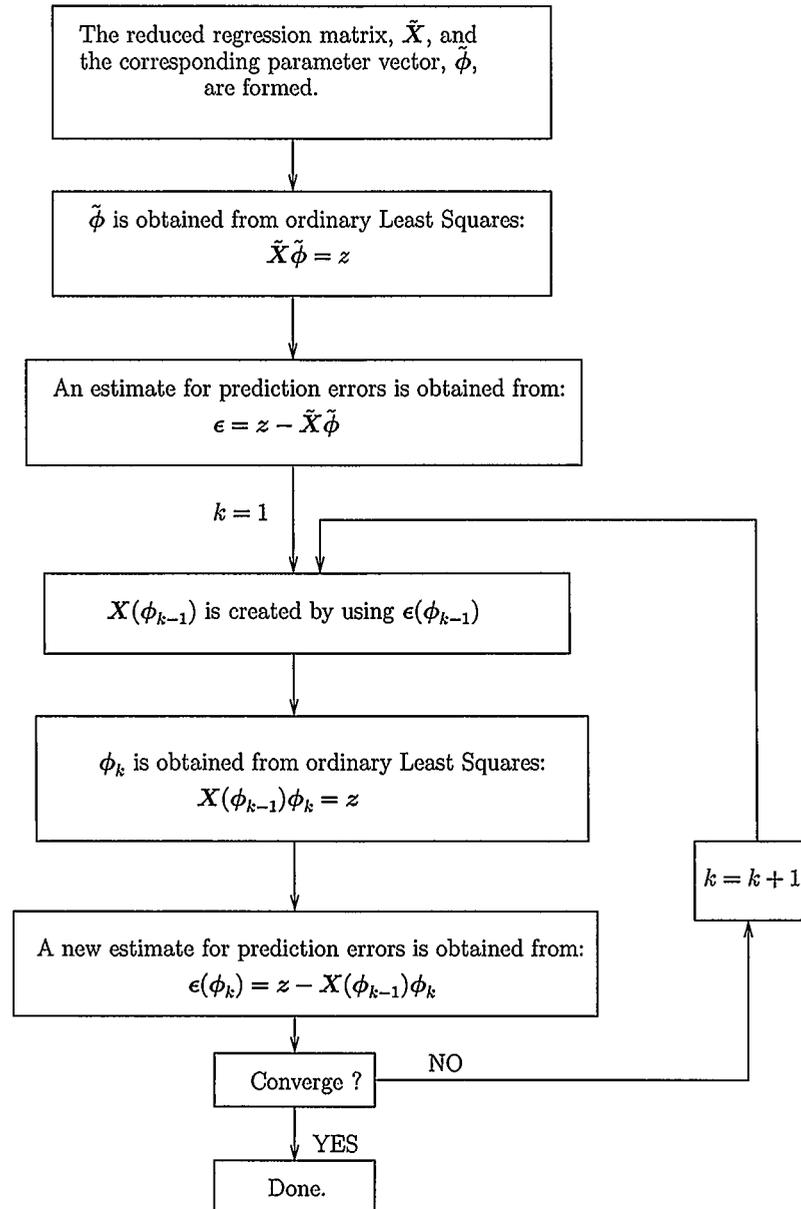


Figure 2.5: Summary of the Extended Least Squares technique

in-the-parameters. If a system model does not fit into this category, then a different optimization routine has to be utilized to minimize the cost function.

Typically, an iterative optimization routine is used for models that are non-linear in the parameters. The main principle behind iterative optimization is to start with an initial vector of parameter estimates, ϕ_0 , and iteratively update the estimates according to:

$$\phi_{k+1} = \phi_k + \mu_k \mathbf{d}_k \quad (2.32)$$

where μ_k is the k^{th} step size and \mathbf{d}_k is the k^{th} direction vector. The step size and the direction vector are chosen such that:

$$V_N(\phi_{k+1}) \leq V_N(\phi_k) \quad (2.33)$$

There are a large number of potential values for the search direction and step size that could satisfy the condition in (2.33). One of the simplest, and most popular, approaches is to update the parameter vector in the direction of *steepest descent* (of the cost function), i.e.:

$$\mathbf{d}_k = -\frac{\partial V_N(\boldsymbol{\phi})}{\partial \boldsymbol{\phi}} \quad (2.34)$$

$$\begin{aligned} &= -\frac{\partial}{\partial \boldsymbol{\phi}} \left(\frac{1}{2N} \sum_{t=1}^N (z(t) - \hat{z}(t, \boldsymbol{\phi}))^2 \right) \\ &= \frac{1}{N} \sum_{t=1}^N (z(t) - \hat{z}(t, \boldsymbol{\phi})) \left(\frac{\partial \hat{z}(t, \boldsymbol{\phi})}{\partial \boldsymbol{\phi}} \right) \\ &= \frac{1}{N} \mathbf{J}^T \boldsymbol{\epsilon} \end{aligned} \quad (2.35)$$

where \mathbf{J} is the Jacobian matrix, i.e. the partial derivatives of the estimated outputs with respect to the parameters. For example, in a model with m parameters and N measurements, the Jacobian is defined as the following $N \times m$ matrix:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \hat{z}(t=0, \boldsymbol{\phi})}{\partial \phi(1)} & \dots & \frac{\partial \hat{z}(t=0, \boldsymbol{\phi})}{\partial \phi(m)} \\ \vdots & \ddots & \vdots \\ \frac{\partial \hat{z}(t=N, \boldsymbol{\phi})}{\partial \phi(1)} & \dots & \frac{\partial \hat{z}(t=N, \boldsymbol{\phi})}{\partial \phi(m)} \end{bmatrix} \quad (2.36)$$

The inherent problem with gradient descent optimization is that the narrowest curvature on the error surface controls the step size, while the most gradual curvature controls the convergence rate [43]. To overcome this problem, the step size needs to be increased in the directions with low curvature and decreased in the directions with high curvature. As a result, the curvature of the error surface (the second derivative with respect to the parameters) needs to be used to modify the step direction. Using the surface curvature to adjust the step direction can also be justified mathematically: performing the Taylor Series expansion on the error surface, about the point $\boldsymbol{\phi}_k$ (parameter values at the k^{th} iteration) gives:

$$V_N(\boldsymbol{\phi}_k + \boldsymbol{\delta}) = V_N(\boldsymbol{\phi}_k) + \left(\frac{\partial V_N}{\partial \boldsymbol{\phi}} \Big|_{\boldsymbol{\phi}=\boldsymbol{\phi}_k} \right)^T \boldsymbol{\delta} + \frac{1}{2!} \boldsymbol{\delta}^T \left(\frac{\partial^2 V_N}{\partial \boldsymbol{\phi}^2} \Big|_{\boldsymbol{\phi}=\boldsymbol{\phi}_k} \right) \boldsymbol{\delta} + \dots \quad (2.37)$$

The Hessian is now defined as a matrix containing the second partial derivatives of the error surface with respect to the parameters:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 V_N(\boldsymbol{\phi})}{\partial \phi(1)^2} & \dots & \frac{\partial^2 V_N(\boldsymbol{\phi})}{\partial \phi(1) \partial \phi(m)} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 V_N(\boldsymbol{\phi})}{\partial \phi(m) \partial \phi(1)} & \dots & \frac{\partial^2 V_N(\boldsymbol{\phi})}{\partial \phi(m)^2} \end{bmatrix} \quad (2.38)$$

Removing the higher order terms in (2.37) and using (2.35) and (2.38) results in:

$$V_N(\boldsymbol{\phi}_k + \boldsymbol{\delta}) \simeq V_N(\boldsymbol{\phi}_k) - \frac{1}{N} (\boldsymbol{\epsilon}^T \mathbf{J}) \boldsymbol{\delta} + \frac{1}{2!} \boldsymbol{\delta}^T \mathbf{H} \boldsymbol{\delta} \quad (2.39)$$

Minimizing the expression for V_N in terms of $\boldsymbol{\delta}$ would produce:

$$\begin{aligned} 0 &= \frac{-1}{N} \mathbf{J}^T \boldsymbol{\epsilon} + \mathbf{H} \boldsymbol{\delta} \\ \therefore \boldsymbol{\delta} &= \frac{1}{N} \mathbf{H}^{-1} \mathbf{J}^T \boldsymbol{\epsilon} \end{aligned} \quad (2.40)$$

which is the gradient descent step of (2.35) with the addition of the Hessian to account for curvature of the error surface. This new step direction is called the *Newton Step*.

Considering one term of the Hessian matrix, more specifically, entry(j,i) in \mathbf{H} as

defined in (2.38):

$$\begin{aligned} \mathbf{H}(j, i) &= \frac{\partial^2 V_N(\boldsymbol{\phi})}{\partial \phi(j) \partial \phi(i)} \\ &= \frac{\partial}{\partial \phi(j)} \left(\frac{\partial V_N(\boldsymbol{\phi})}{\partial \phi(i)} \right) \end{aligned} \quad (2.41)$$

where $\frac{\partial V_N(\boldsymbol{\phi})}{\partial \phi(i)}$ can be obtained by using (2.35), so that:

$$\begin{aligned} \mathbf{H}(j, i) &= \frac{\partial}{\partial \phi(j)} \left(\frac{-1}{N} \mathbf{J}^T(:, i) \boldsymbol{\epsilon} \right) \\ &= \frac{1}{N} \mathbf{J}^T(:, i) \mathbf{J}^T(:, j) - \frac{1}{N} \left(\frac{\partial^2 \hat{z}(\boldsymbol{\phi})}{\partial \phi(j) \partial \phi(i)} \right)^T \boldsymbol{\epsilon} \end{aligned} \quad (2.42)$$

The second term in (2.42) is computationally expensive (due to the second order derivatives) and will be insignificant close the optimal point (since the prediction error, $\boldsymbol{\epsilon}$, will be very small). As a result, the *Gauss-Newton* Method uses the first term in (2.42) to obtain an approximation to the Hessian:

$$\hat{\mathbf{H}} = \frac{1}{N} \mathbf{J}^T \mathbf{J} \quad (2.43)$$

which converts the Newton step from (2.5.3) to:

$$\mathbf{d}_k = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \boldsymbol{\epsilon} \quad (2.44)$$

resulting in the following parameter update method:

$$\phi_{k+1} = \phi_k + \mu_k (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \epsilon \quad (2.45)$$

where μ_k is chosen adaptively.

The main problem in the Gauss-Newton optimization routine is that the approximate Hessian of (2.43) can become singular or poorly-conditioned [43]. To overcome this problem, the *Levenberg-Marquardt* (LM) approach is introduced, which adds a diagonal regularization term to the approximate Hessian of the Gauss-Newton approach [40]. Therefore, the approximate Hessian for the LM approach is defined as:

$$\hat{\mathbf{H}} = \frac{1}{N} (\mathbf{J}^T \mathbf{J} + \mu_k \mathbf{I}) \quad (2.46)$$

where μ_k is the size of the diagonal ridge and \mathbf{I} is the Identity matrix. The parameter update vector is therefore given by:

$$\phi_{k+1} = \phi_k + (\mathbf{J}^T \mathbf{J} + \mu_k \mathbf{I})^{-1} \mathbf{J}^T \epsilon \quad (2.47)$$

The ridge size in the LM algorithm, μ_k , is chosen adaptively, based on the distance from the optimal solution. The method presented in [40] is used in this work to choose μ_k (μ_k is halved after a successful attempt, i.e. a reduction in the cost function, and doubled after a failed attempt).

2.6 Practical Applications

In this section, two models that will be studied in this document are presented: a power system dynamic non-linear load model and a parallel pathway model of ankle dynamics.

2.6.1 Power System Load Models

One of the standard dynamic load models, which has been shown to accurately represent the steady-state non-linear behavior of the load along with load recovery and overshoot [25], and has been recommended by the IEEE Power System Stability Subcommittee [12], is defined as follows :

$$\dot{x}(t) = -\frac{x(t)}{T_p} + P_o \left[\frac{V_L(t)}{V_o} \right]^{N_{ps}} - P_o \left[\frac{V_L(t)}{V_o} \right]^{N_{pt}} \quad (2.48a)$$

$$P_d(t) = \frac{x(t)}{T_p} + P_o \left[\frac{V_L(t)}{V_o} \right]^{N_{pt}} \quad (2.48b)$$

$$\dot{z}(t) = -\frac{z(t)}{T_q} + Q_o \left[\frac{V_L(t)}{V_o} \right]^{N_{qs}} - Q_o \left[\frac{V_L(t)}{V_o} \right]^{N_{qt}} \quad (2.49a)$$

$$Q_d(t) = \frac{z(t)}{T_q} + Q_o \left[\frac{V_L(t)}{V_o} \right]^{N_{qt}} \quad (2.49b)$$

where P_d and Q_d are, respectively, the active and reactive power demand of the load at time t , $x(t)$ and $z(t)$ are internal state variables, T_p and T_q are the time constants for $x(t)$ and $z(t)$, N_{ps} and N_{pt} are the steady-state and transient voltage indices for active power, and N_{qs} and N_{qt} are the steady-state and transient voltage indices for reactive power, respectively. It is assumed that the nominal values, P_o and V_o are

known. Therefore, the load voltages, $V_L(t)$, can be normalized as $V(t) := \frac{V_L(t)}{V_o}$.

For the remainder of this document, only the active power load model will be studied. It can be seen from (2.48) and (2.49) that the reactive power model has the same structure as the active power component. Therefore, the same algorithm that is used for identifying the active power model can be used to identify the reactive power as well.

Using standard system identification nomenclature, and assuming the nominal values are known, the parameters of interest in the active power model (2.48) are:

$$\theta = [N_{ps} \ N_{pt} \ T_p^{-1}] = [\theta_1 \ \theta_2 \ \theta_3] \quad (2.50)$$

Using the notation in (2.50) and defining $y(t) = P_d(t)$, (2.48) can be re-written as:

$$\begin{aligned} \dot{x}(t) &= -\theta_3 x(t) + P_o V^{\theta_1}(t) - P_o V^{\theta_2}(t) \\ y(t) &= \theta_3 x(t) + P_o V^{\theta_2}(t) \end{aligned} \quad (2.51)$$

Introducing the following algebraic simplification:

$$w_1(t) := f_1(V) := P_o V^{\theta_1}(t) - P_o V^{\theta_2}(t) \quad (2.52a)$$

$$w_2(t) := f_2(V) := P_o V^{\theta_2}(t) \quad (2.52b)$$

allows (2.51) to be re-written as:

$$\dot{x}(t) = -\theta_3 x(t) + w_1(t) \quad (2.53a)$$

$$y(t) = \theta_3 x(t) + w_2(t) \quad (2.53b)$$

Since digital computers are used in all of the data processing, and since the measurements are taken at discrete time intervals, the continuous-time model of (2.53) needs to be recast in discrete-time. In order to discretize the system using a zero order hold (ZOH) [54], the system input must remain constant between samples (i.e. either if the sampling rate is very high compared to the system dynamics, or when the system input is provided by a digital controller [39]). For the load model considered in this document, the sampling rate is high relative to the system dynamics. Using the ZOH assumption, the continuous time model (2.53) can be discretized as follows [54]:

$$x(k\ell + \ell) = e^{-\theta_3 \ell} x(k\ell) + \frac{1 - e^{-\theta_3 \ell}}{\theta_3} w_1(k\ell) \quad (2.54a)$$

$$y(k\ell) = \theta_3 x(k\ell) + w_2(k\ell) \quad (2.54b)$$

where k is the time index and ℓ is the sampling period.

The following simple manipulations are done to write the output, $y(\cdot)$, directly in terms of the input $u(\cdot)$, i.e., to eliminate the internal state variable $x(\cdot)$. Using the forward time shift operator q , (2.54a) can be written as:

$$qx(k\ell) = e^{-\theta_3\ell}x(k\ell) + \frac{1 - e^{-\theta_3\ell}}{\theta_3}w_1(k\ell)$$

which can be further simplified to:

$$(q - e^{-\theta_3\ell})x(k\ell) = \frac{1 - e^{-\theta_3\ell}}{\theta_3}w_1(k\ell)$$

Isolating $x(k\ell)$ gives:

$$x(k\ell) = \frac{1 - e^{-\theta_3\ell}}{\theta_3(q - e^{-\theta_3\ell})}w_1(k\ell) \quad (2.55)$$

Finally, substituting (2.55) in (2.54b) produces:

$$y(k\ell) = \frac{1 - e^{-\theta_3\ell}}{q - e^{-\theta_3\ell}}w_1(k\ell) + w_2(k\ell) \quad (2.56)$$

The discrete-time dynamic load model of (2.56) was derived in [33] where the authors used an Adaptive Simulated Annealing approach to estimate the model parameters.

For the algorithms presented in this document, it is advantageous to represent the block structure version of (2.56), which is shown in Figure 2.6 (with $w_1(k\ell)$ and $w_2(k\ell)$ defined by (2.52)).

The lower branch in the model represents the second term in the right hand side (RHS) of (2.56), $w_2(k\ell)$, which is a memoryless non-linearity applied to the input. More specifically, the output of this branch is an exponential in the instantaneous value of the input ($f_2(V) = P_0V^{\theta_2}$).

The top branch of the model represents the first term in the RHS of (2.56).

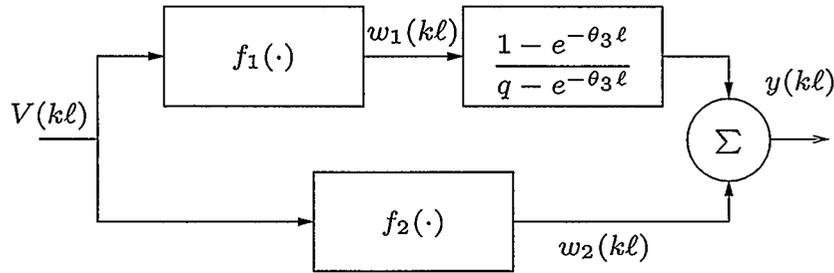


Figure 2.6: The block structure version of the discrete-time dynamic load model

This branch consists of two components: the first element is another memoryless non-linearity on the input,

$$w_1(kl) = f_1(V) = P_0 V^{\theta_1} - P_0 V^{\theta_2}.$$

The second part of the branch, $\frac{1 - e^{-\theta_3 l}}{q - e^{-\theta_3 l}}$, is a first order filter, with a gain of $1 - e^{-\theta_3 l}$ and a pole at $e^{-\theta_3 l}$. This branch is a Hammerstein structure

To account for measurement errors, independent Gaussian noise, $e(t)$, is added to the output signal, resulting in the following input-output relationship:

$$z(kl) = \frac{1 - e^{-\theta_3 l}}{q - e^{-\theta_3 l}} w_1(kl) + w_2(kl) + e(kl) \quad (2.57)$$

where z is the contaminated signal, i.e., $z(kl) = y(kl) + e(kl)$.

2.6.2 Parallel Pathway Model of Ankle Dynamics

The mechanical behaviour of a joint is defined by the relationship between the position of that joint and the torque acting about it [41]. In other words, this relationship indicates how a given joint interacts with its associated limbs and its en-

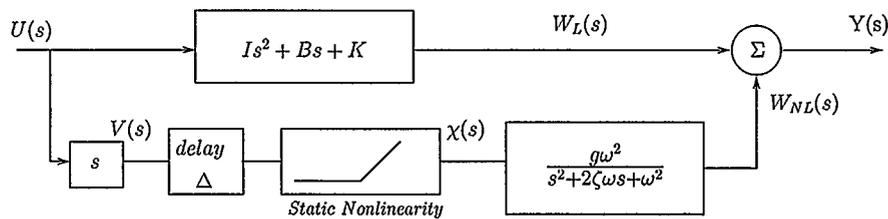


Figure 2.7: Continuous-Time model of ankle dynamics showing the relationship between the angle and the net torque of the ankle

environment [31, 41]. The relationship can be categorized into two main components: intrinsic and reflex. The intrinsic component is based on the mechanical properties of the joint, passive tissue, and active muscle fiber. The reflex component is caused by the sensory response to stretch. [41]

A parallel pathway model for the ankle, developed by the researchers at Neuro-muscular Control Laboratory at McGill University, is represented in Figure 2.7 [32]. This model represents the relationship between the position (angle) of the ankle, $U(s)$, and the net ankle torque, $Y(s)$. The block-structured model is divided into two components: the linear (upper) path and the non-linear component (lower path).

In the upper path, the intrinsic stiffness of the ankle is represented as a second order system, resulting in the linear contribution to the ankle torque, $W_L(s)$. In this second order system, I , B and K represent inertia, viscosity, and elasticity of the ankle joint, respectively.

In the lower path, the reflex stiffness of the ankle is represented as a group of four components, resulting in the non-linear contribution to the ankle torque, $W_{NL}(s)$. The components of this path, in order, are: a derivative block, which converts the ankle angle $U(s)$ to ankle velocity $V(s)$, a delay block, a static non-linearity, and a

low-pass filter which models muscle activation.

The static non-linearity is a half-wave rectifier, with the following definition:

$$\chi(t) = \begin{cases} 0, & v(t - \Delta) \leq 0 \\ v(t - \Delta), & v(t - \Delta) > 0 \end{cases} \quad (2.58)$$

The muscle activation filter includes the following parameters: g , ζ , and ω , which represent the filter gain, damping parameter, and natural frequency [32].

The ankle dynamics model can now be summarized as:

$$Y(s) = W_L(s) + W_{NL}(s) \quad (2.59)$$

given:

$$\begin{aligned} W_L(s) &= (Is^2 + Bs + K) \cdot U(s) \\ W_{NL}(s) &= \frac{g\omega^2}{s^2 + 2\zeta\omega s + \omega^2} e^{-s\Delta} X(s) \end{aligned} \quad (2.60)$$

where $X(s)$ is the rectified version, (2.58), of the ankle velocity $V(s)$.

The proposed algorithm, described in Section 3.4, will introduce mathematical formulation and the discrete-time conversion for the ankle dynamics.

2.7 Summary

In this chapter, the problem of system identification is formulated in a mathematical framework. Then, Hammerstein and NARMAX models are presented and dis-

cussed. A few optimization routines, namely Ordinary Least Squares, Extended Least Squares, and Levenberg-Marquardt, are also introduced. These routines will be utilized in the proposed algorithms in Chapters 3 and 4. Finally, the power system load model and the ankle dynamic model are presented.

Chapter 3

Multi-stage System Identification

3.1 Introduction

Once an appropriate system model structure has been established, the next step in the system identification process is estimating the model parameters. If the estimation problem is non-convex, finding the optimal parameters can be a challenging task due to the presence of local minima in the solution space. A *global search* of the solution space for finding the optimal parameters is very time-consuming, and the quality of a *local search* is dependent on the initial values. In this chapter, a new, multi-stage algorithm is proposed to help overcome the difficulties associated with non-convex system identification. The proposed algorithm is then applied to the problem of power system load identification and ankle dynamics identification in order to examine its validity.

3.2 Non-convex System Identification

The main principle of the proposed algorithm is to replace the non-convex identification problem with an approximate, but convex, counterpart. The parameters from the approximate model are then used as initial values for finding the optimal parameters of the original, non-convex problem.

The main steps in the proposed algorithm are as follows:

1. The non-linear system is approximated as a linear-in-the-parameters model: (2.18) or (2.28).
2. Ordinary Least Squares or Extended Least Squares are used to identify the optimal parameters of the model introduced in Step 1.
3. The optimal parameters from Step 2 are mapped to the parameters of the original non-linear system.
4. A local optimization routine is performed on the non-linear system to find its optimal parameters. The values found in Step 3 are used as initial values for the optimization process.

3.3 Power System Load Model Identification

Multi-stage system identification is used to overcome difficulties associated with non-linear and/or non-convex problems. The output of the discrete-time non-linear model (2.57) is a non-linear function of the model parameters $(\theta_1, \theta_2, \theta_3)$ and the resulting optimization is non-convex. Therefore, a multi-stage approach is proposed, where the initial stages are utilized to determine starting values for the non-linear optimization routine that determines the optimal physical parameters.

3.3.1 Proposed Multi-stage Identification Algorithm

Using the approach of Section 3.2, the main steps for identifying the power system load model are:

1. The memoryless non-linearities in the discrete-time model are approximated with polynomials to ease the parameter estimation process (NARMAX formulation).
2. Input and output measurements are used to determine the parameters of the NARMAX approximation.
3. The NARMAX parameters are mapped to estimated physical (actual) parameters of the aggregate load model
4. The estimated physical parameters are used as initial values in a local optimization routine (Levenberg-Marquardt algorithm). The goal of the routine is to minimize the difference between the measured output and that predicted by the model.

The remainder of this section gives details of all the stages in the proposed algorithm.

Stage 1: NARMAX formulation of the proposed model

In the first stage, the non-linear and non-convex physical load model is approximated with a linear-in-the-variables model. The parameter values for the approximated model can be found using a pseudo-linear regression approach [9, 22]. Specifically, a NARMAX formulation, which is linear-in-the-variables, is introduced as an approximation to the discrete-time model of (2.57) [37, 38].

To utilize a NARMAX formulation, the exponential terms in the model need to be approximated with a basis expansion [38]. For this thesis, 2^{nd} order polynomials are chosen to approximate the input non-linearities (this approach will be justified

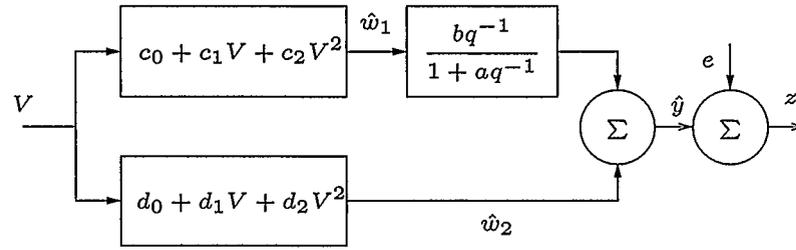


Figure 3.1: The polynomial approximation to the discrete-time dynamic load model, used to derive the NARMAX representation.

in Section 3.3.4. Other expansions were not considered because of difficulties that would arise in Stage 3 when mapping from the basis parameters to the physical parameters.) More specifically, $w_1(k\ell)$ and $w_2(k\ell)$ are approximated as:

$$\hat{w}_1(t) = c_0 + c_1 V + c_2 V^2$$

$$\hat{w}_2(t) = d_0 + d_1 V + d_2 V^2$$

Throughout the remainder of this section, the $\hat{\cdot}$ notation is used to indicate estimates, and t is implied to be in discrete time.

The resulting block structured model is shown in Figure 3.1, where b represents the gain of the 1st order filter, a is the pole location for the filter, and e is the white Gaussian noise added to the output.

From Figure 3.1, the output $z(\cdot)$ is written as the summation of the signal of the upper branch ($\hat{w}_1(t)$ after the first order filter), the signal of the lower branch ($\hat{w}_2(t)$) and the white Gaussian noise e . Mathematically, this can be written as:

$$z(t) = ((a + 1)d_o + bc_o) + d_1V(t) + d_2V^2(t) + (bc_1 + ad_1)V(t - 1) + (bc_2 + ad_2)V^2(t - 1) - az(t - 1) + ae(t - 1) + e(t) \quad (3.1)$$

which is pseudo-linear in the parameters and is a NARMAX formulation [37, 38].

The vector of parameters to be identified is ψ , which contains the polynomial coefficients and filter coefficients:

$$\psi = [a \ b \ c_0 \ c_1 \ c_2 \ d_0 \ d_1 \ d_2]$$

Stage 2: NARMAX parameter estimation

To simplify the identification process, the model of (3.1) is rewritten in the pseudo-linear form. Similar to (2.27), the error terms, $e(t)$, in the regressor matrix are replaced with prediction errors, $\epsilon(t)$, since the error terms can not be measured. The pseudo-linear input-output relationship is:

$$z(t) = \mathbf{x}(t, \phi)^T \phi + e(t) \quad (3.2)$$

where

$$\mathbf{x}(t, \phi) = \begin{bmatrix} 1 \\ V(t) \\ V^2(t) \\ V(t-1) \\ V^2(t-1) \\ z(t-1) \\ \epsilon(t-1, \phi) \end{bmatrix} \quad \phi = \begin{bmatrix} (a+1)d_o + bc_o \\ d_1 \\ d_2 \\ bc_1 + ad_1 \\ bc_2 + ad_2 \\ -a \\ a \end{bmatrix} \quad (3.3)$$

For a set of n measurements, the regressor matrix, \mathbf{X} , is defined as:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(t_1, \phi)^T \\ \mathbf{x}(t_2, \phi)^T \\ \vdots \\ \mathbf{x}(t_n, \phi)^T \end{bmatrix} \quad (3.4)$$

Since $\mathbf{x}(t, \phi)$ depends on the model parameters, due to the inclusion of $\epsilon(t-1, \phi)$, the model is not strictly linear in terms of all the parameters. Thus, an Ordinary Least Squares algorithm can not be used to determine the parameter values from input and output measurements. To overcome this difficulty, an Extended Least Squares (ELS) algorithm [62] can be used to estimate the parameters of the NARMAX model, ϕ [39]. The Extended Least Squares method is presented in Section 2.2.

With $\hat{\phi}$ obtained using ELS, ϕ from (3.3) is used to compute estimates of the parameter values, (the elements of the vector ψ), as follows:

$$\begin{aligned}
\hat{d}_1 &= \hat{\phi}[2] \\
\hat{d}_2 &= \hat{\phi}[3] \\
\hat{a} &= -\hat{\phi}[6] \\
\hat{b} &= 1 + \hat{a} \\
\hat{c}_1 &= \frac{\hat{\phi}[4] - \hat{a}\hat{d}_1}{1 + \hat{a}} \\
\hat{c}_2 &= \frac{\hat{\phi}[5] - \hat{a}\hat{d}_2}{1 + \hat{a}}
\end{aligned}$$

where the brackets indicate the entry of the vector ϕ .

The only parameters that can not be explicitly determined are c_0 and d_0 , both of which only appear in the value of $\phi[1]$. They are therefore estimated in the next stage as part of the mapping algorithm.

Stage 3: Mapping between polynomial coefficients and physical parameters

Using a least squares curve-fitting approach [17], the values obtained for the polynomial coefficients, $\hat{\psi}$, are mapped into initial estimates of the physical parameters, θ .

The relationship between the first order filter associated with the physical model and the NARMAX approximated system can be used to form an initial estimate of θ_3 . In the physical model of Figure 2.6, the first order filter is defined as $\frac{1-e^{-\theta_3 \ell}}{q-e^{-\theta_3 \ell}}$; in the NARMAX model, the first order filter term:

$$\frac{bq^{-1}}{1 + aq^{-1}}$$

can be rewritten as:

$$\frac{b}{q + a} \tag{3.5}$$

Equating the denominator of the physical system and (3.5) gives:

$$\begin{aligned} q + a &= q - e^{-\theta_3 \ell} \\ \Rightarrow a &= -e^{-\theta_3 \ell} \\ \hat{\theta}_3 &= \frac{-\log(-\hat{a})}{\ell} \end{aligned}$$

where log represents the natural logarithm (base e).

Next, the terms in $\hat{\psi}$ determined in Stage 2 are used to estimate θ_1 , θ_2 , c_0 , and d_0 using a curve-fitting approach. The steps in this process are as follows:

1. Since the polynomial $d_o + d_1V + d_2V^2$ was used in the lower branch of Figure 3.1 to approximate V^{θ_2} , $V^{\theta_2} - d_o$ is fitted to $\hat{d}_1V + \hat{d}_2V^2$ using a non-linear least squares algorithm to obtain the estimates of θ_2 and d_o .
2. Using the value of \hat{d}_o calculated in the above step (and the values of \hat{a} and \hat{b} determined in Stage 2), the value of c_0 is determined using the first element of vector ϕ from (3.3), i.e.

$$\hat{\phi}[1] = (a_0 + 1)d_0 + bc_0 \tag{3.6}$$

3. In the upper branch of Figure 3.1, the polynomial $c_o + c_1V + c_2V^2$ was used to estimate $V^{\theta_1} - V^{\theta_2}$. Since θ_2 and the coefficients c_o , c_1 , and c_2 have been estimated, $\hat{c}_o + \hat{c}_1V + \hat{c}_2V^2 + V^{\hat{\theta}_2}$ is fitted to V^{θ_1} in order to obtain an estimate for θ_1 .

Initial estimates of all the physical parameters have now been determined.

Stage 4: Levenberg-Marquardt Algorithm

In the final stage, the estimates of the physical parameters $\hat{\theta} = [\hat{\theta}_1 \ \hat{\theta}_2 \ \hat{\theta}_3]$, found via Stages 1 through 3, are used as initial estimates for a Levenberg-Marquardt routine that determines their optimal values.

To perform the optimization routine, the equations for the Jacobian need to be derived. One row of the Jacobian is defined as:

$$J(t, :) = \left[\frac{\partial \hat{y}(t)}{\partial \theta_1} \quad \frac{\partial \hat{y}(t)}{\partial \theta_2} \quad \frac{\partial \hat{y}(t)}{\partial \theta_3} \right]$$

To derive the Jacobian terms, the expression for $\hat{y}(t, \theta)$ from (2.57) is used to determine the derivatives:

$$\begin{aligned} \frac{\partial \hat{y}(t, \theta)}{\partial \theta_1} &= \frac{e^{-\theta_3 \ell} - 1}{q - e^{-\theta_3 \ell}} P_o V^{\theta_1}(t) \log V(t) \\ \frac{\partial \hat{y}(t, \theta)}{\partial \theta_2} &= \frac{1 - q}{q - e^{-\theta_3 \ell}} P_o V^{\theta_2}(t) \log V(t) \\ \frac{\partial \hat{y}(t, \theta)}{\partial \theta_3} &= \ell e^{-\theta_3 \ell} \frac{1 - q}{(q - e^{-\theta_3 \ell})^2} P_o \left(V^{\theta_1}(t) - V^{\theta_2}(t) \right) \end{aligned} \quad (3.7)$$

A flow chart summarizing all the stages in the multi-stage identification process for the dynamic load model is shown in Figure 3.2. It should be noted that this algo-

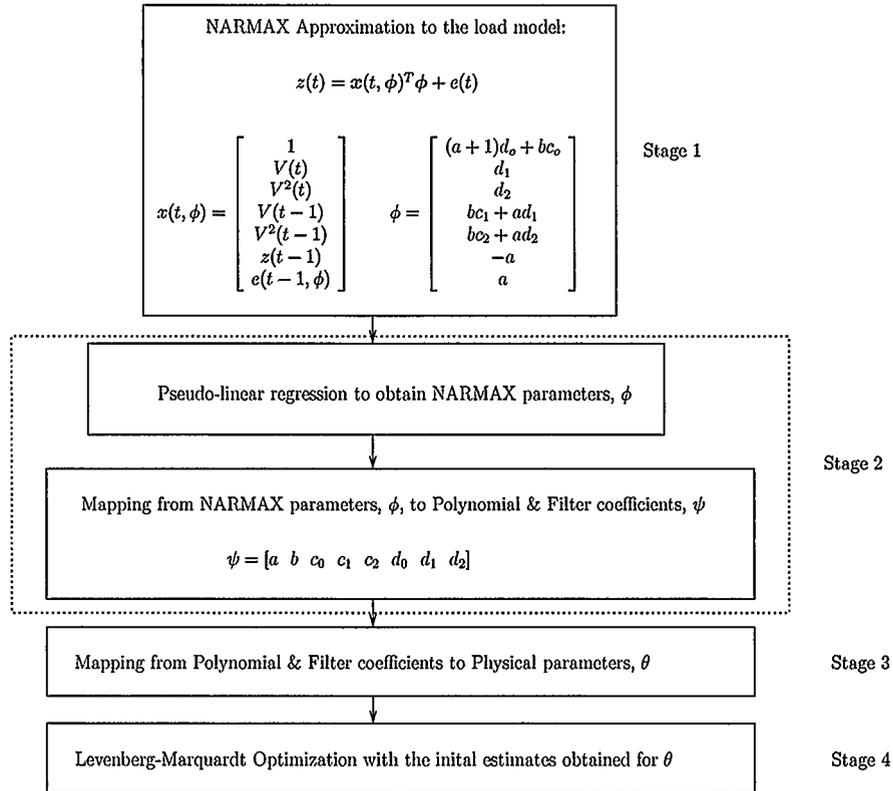


Figure 3.2: Summary of the multi-stage identification process for the non-linear dynamic load model

rithm can not *guarantee* convergence to the global optimum. However, as shown in the following section, it converges to the global optimum for the problems considered in the thesis.

3.3.2 Simulated System Results

To study the proposed approach, a simulated system is used. The parameters of the model in (2.57), $[\theta_1 \ \theta_2 \ \theta_3]$, are set to $[1.2 \ 1.7 \ 0.4]$ (Based on the parameters given in [33].) The input signal, V , has a uniform distribution between 0.9 and 1.1, which is reasonable since those values are the typical limits on per-unit voltage. The

	Sample Mean		Sample Std Dev ($\times 10^{-2}$)		actual value
	ASA	Multi-Stage	ASA	Multi-Stage	
θ_1	1.222	1.205	7.77	5.70	1.2
θ_2	1.7040	1.695	2.02	2.04	1.7
θ_3	0.3611	0.421	6.27	7.17	0.4

Table 3.1: Parameter statistics for ASA and the Multi-stage approach from 30 Monte Carlo runs with $N = 1000$ points.

additive white Gaussian noise is set to be $\mathcal{N}(0, 0.0015)$ – Normal distribution with 0 mean and variance of 0.0015 – for an SNR of approximately 25 dB. The values for the sampling period (ℓ), nominal voltage (V_o), and nominal power (P_o) are set to 1 second, 1 volt, and 1 watt, respectively.

To analyze the time performance of the proposed algorithm, the Adaptive Simulated Annealing (ASA) technique [26], used in [33], was also implemented. The ASA technique is a global search mechanism that is statistically guaranteed to find the optimal solution if it is given infinite time.

The same initial parameter set as [33], [0.5 2.0 1.7], is used as the starting point for the ASA technique.

Table 3.1 demonstrates the parameter statistics when 30 Monte Carlo runs are used on an artificial data set containing 1000 points. The sample mean and standard deviation from both approaches (ASA and multi-stage) are presented in the table. It can be seen that both techniques can successfully estimate the parameters of the simulated system. The parameters obtained from ASA have an error percentage less than 9.8% and those from the proposed algorithm fall within 5.3% of the simulated values. Also, all simulated values are within one standard deviation of the estimated parameters.

The main difference between the two approaches is in their respective time performances. As expected, ASA is very time consuming since it requires extensive searching. Thus, the proposed algorithm presents a major advantage over ASA in terms of computational burden. To obtain the parameters reported in Table 3.1, ASA required 1897.5 seconds while the multi-stage identification technique required only 13.8 seconds, under identical conditions (on an AMD Athlon based PC).

The proposed algorithm was further tested on different sets of theoretical parameters: [1.5 2.0 0.2], [2.0 1.5 0.5], [2.1 1.8 0.4], [1.0 1.7 0.3], [1.5 1.6 1.1], and [1.2 2.7 0.3448] were used as theoretical θ values. In all cases, the simulated values were within one standard deviation of the estimated parameters, the mean of the estimated parameters had an error percentage less than 6.8% for the proposed algorithm and less than 10.1% for the ASA approach, and the proposed approach was – on average – more than 14 times faster than ASA.

To examine the proposed algorithm, an identification-validation test is performed using simulated data. A set of 8000 data points is created using the same parameters as those introduced at the beginning of Section 3.3.2. The data set is split into two components: The first 5000 data points are used in the identification process (The proposed algorithm is applied to these data in order to estimate the parameters). The second 3000 data points are used in the validation process (The estimated parameters from the identification process are used to construct the predicted output for this set of data). The residuals are calculated for both data sets: ϵ_{id} for the identification set, and ϵ_{va} for the validation set. The mean squared values of the residuals are:

$$\frac{1}{N} \sum \epsilon_{id}^2 = 1.5329 \times 10^{-3} \quad \frac{1}{N} \sum \epsilon_{va}^2 = 1.5010 \times 10^{-3}$$

Since the errors for the two sets are very similar, it can be concluded that the proposed algorithm is successful in accurately predicting the model parameters.

3.3.3 Asymptotic Behaviour

In this section, the asymptotic behaviour of the parameter estimates is investigated to verify that the proposed algorithm produces consistent estimates. The objective was to determine whether or not

$$\lim_{N \rightarrow \infty} \text{var } \hat{\boldsymbol{\theta}}(N) = 0$$

In other words, do the estimated parameters converge (to the optimal values) as the number of data samples increases? In particular, the Cramer-Rao Lower Bound (CRLB) [39] is used to assess the quality of the estimated parameters and consequently, the quality of the identification. The CRLB is the theoretical lower bound on the variance attainable for unbiased parameter estimates; the closer the variance of the estimates are to the CRLB, the more consistent the parameters are [39]. The CRLB is computed using the inverse of the Fisher Information Matrix, \mathbf{M} . For independent, identically distributed (*iid*) Gaussian noise [47], \mathbf{M} is defined as [39]:

$$\mathbf{M} = \frac{1}{\sigma_e^2} (\mathbf{J}^T \mathbf{J})$$

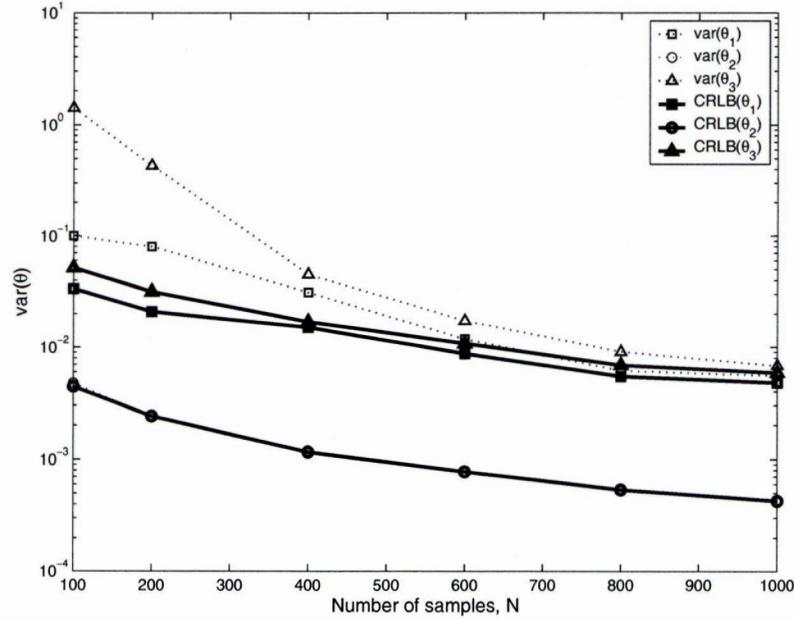


Figure 3.3: The variance (var) and the Cramer-Rao Lower bound (CRLB) of the parameter estimates ($\theta_1, \theta_2, \theta_3$) obtained from 5000 Monte Carlo runs of the proposed approach.

where σ_e^2 is the noise variance and \mathbf{J} is the Jacobian matrix of the estimated outputs, as defined in (3.7)

Figure 3.3 shows the variances of the estimated parameters as the number of samples, N , is increased from 100 to 1000. The Cramer-Rao Lower Bound for each experiment is also shown on the graph. The estimates are obtained from 5000 Monte Carlo simulations at each value of N .

For $N > 400$, the parameter variances are close to the Cramer-Rao Lower Bound, and the clear trend is that the variances show asymptotic behavior and will continue to approach the CRLB.

3.3.4 Model Structure Verification

In order to examine the validity of the approximations made during Stage 1 of the proposed algorithm, a number of experiments were performed. The goals of these experiments were twofold:

1. To examine the validity of approximating V^θ with a 2^{nd} order polynomial
2. To ensure that the NARMAX model of Stage 2 contains all the terms required to represent the load model, i.e. no terms were left behind due to the approximation process.

Verification of polynomial Approximation

To convert the discrete-time non-linear model of (2.57) into a pseudo-linear regression model (Stage 2), the exponential terms were replaced with more suitable functions, specifically second order polynomials. However, the order of these polynomials needs to be verified. Thus, the exponentials in the physical model are approximated by both a 2^{nd} order and a 5^{th} order polynomials.

To confirm the feasibility of using a 2^{nd} order polynomial to approximate the exponential terms, a series of curve-fitting tests were performed on simulated input data, similar to that of Section 3.3.2 (V: uniform [0.9 1.1]). The two exponential terms in the original dynamic load model, V^{θ_2} and $V^{\theta_1} - V^{\theta_2}$ were evaluated within the permissible ranges for θ_1 and θ_2 [1]:

$$\begin{aligned} 0 &\leq \theta_1 \leq 3 \\ 0.5 &\leq \theta_2 \leq 2.5 \end{aligned} \tag{3.8}$$

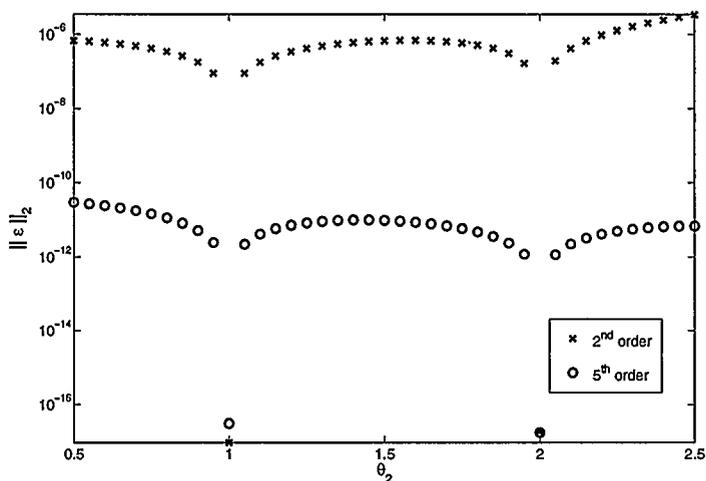


Figure 3.4: The 2-norm of residuals from approximating V^{θ_2} with 2nd and 5th order polynomials

Figure 3.4 shows the 2-norm of the residuals from the curve-fitting process. It can be seen that the 2-norm from the 2nd order polynomial is slightly higher than that of the 5th order polynomial, but the results are acceptable, considering the very small scale of the residuals (10^{-6}). The norm is also bounded for $\theta_2 \leq 2$. The relative size of the error for both cases is very small and irrelevant when the effects of even a small amount of noise are considered.

Results shown in Figure 3.4 indicate the *modeling error* in the approximation process. In other words, the exponential term is assumed to be noise-free. However, in a typical setting, the measurements are contaminated with noise. Therefore, the approximation process is repeated with white Gaussian noise added to the exponential terms. Figure 3.5 shows the 2-norm of the residuals from the curve-fitting process, with a signal-to-noise ratio (SNR) of 30 dB. Figure 3.6 shows the mean and variance of the residuals from the same process. Plots for the $V^{\theta_1} - V^{\theta_2}$ exponential

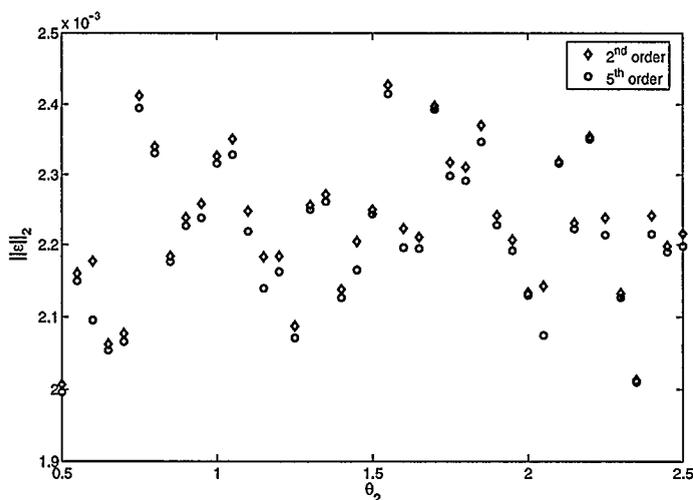


Figure 3.5: The 2-norm of residuals when approximating V^{θ_2} with 2nd and 5th order polynomials (SNR = 30 dB)

term showed similar characteristics.

The above discussion clearly shows that with noise added in the model, a higher order polynomial does not provide any advantages over the 2nd order polynomial approximation.

Verification of the Overall Model Structure

Structure detection, in general, is the process of selecting a subset of candidate parameters that accurately reflect the input-output relationship of a system. Bootstrap is one of the techniques used in the structure detection process.

To demonstrate the concept of structure detection, a simple example, based on the NARMAX model of (2.11), is used: A NARMAX model is to be identified and the user's knowledge of the true system is only limited to the order of the polynomials and the maximum lag on the signals. If the order of the polynomials is 2, ($l = 2$), and

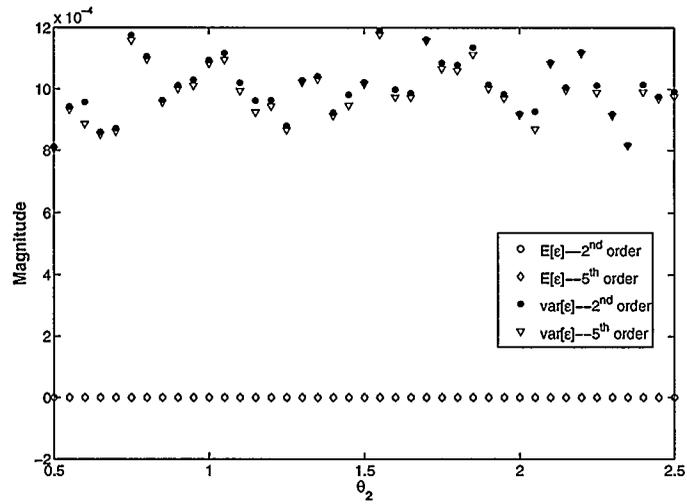


Figure 3.6: Mean and variance of residuals when approximating V^{θ_2} with 2^{nd} and 5^{th} order polynomials (SNR = 30 dB)

the maximum lag on the signals is 1 ($n_u = n_z = n_e = 1$), the *complete* NARMAX model – as previously shown in (2.11) – is:

$$z(t) = \begin{bmatrix} 1 \\ z(t-1) \\ u(t) \\ u(t-1) \\ e(t-1) \\ z(t-1)u(t-1) \\ z(t-1)e(t-1) \\ u(t-1)e(t-1) \\ z^2(t-1) \\ u^2(t) \\ u^2(t-1) \\ e^2(t-1) \\ u(t)u(t-1) \\ u(t)e(t-1) \end{bmatrix}^T \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \end{bmatrix} + e(t) \quad (3.9)$$

However, the true system is likely to contain only a few of the 14 parameters shown in (3.9). If the real system, which is unknown to the user at the start of the structure detection process, is represented by:

$$z(t) = \begin{bmatrix} 1 \\ z(t-1) \\ u(t) \\ u(t-1) \\ z(t-1)e(t-1) \\ u(t-1)e(t-1) \\ u^2(t) \end{bmatrix}^T \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_6 \\ a_7 \\ a_9 \end{bmatrix} + e(t) \quad (3.10)$$

then, parameters a_4 , a_5 , a_8 and $a_{10} - a_{13}$ which exist in the model of (3.9) but are not true parameters of the system, are called *spurious*. In other words, if the identification process is successful, the value of the spurious parameters should be zero. The challenge in the structure detection problem lies in detecting these spurious parameters. One possible approach for solving this problem is to construct the regressor matrix as defined in (3.9), then perform parameter estimation and record the optimal parameters. Repeating this procedure in a Monte Carlo simulation will allow the user to study the distribution of all the parameters and to determine if they are spurious or not. More specifically, if 0 lies in the 99% confidence interval of an estimated parameter, then it is deemed spurious, and can thus be removed from the system model.

In this Section, the load model structure is investigated using the sub-optimal Bootstrap structure detection technique [35]. More specifically, the process of approximating the non-linear load model with a 2^{nd} order NARMAX model is scrutinized. To perform this task, a 2^{nd} order polynomial NARMAX model and a 5^{th} order polynomial NARMAX model are used to identify the non-linear power system

load. Then, the parameters of the proposed NARMAX model from each scenario are examined to determine their significance.

The Bootstrap method is a powerful numerical tool that can be used for computing parameter statistics. The technique requires very few assumptions; namely, that the errors are independent, identically distributed (*iid*) with zero mean. The Bootstrap simulates a Monte Carlo analysis, but it presents a major advantage over that technique since it only requires one set of measurements. In most practical situations, one can only obtain a single (or very few) set of measurements. The Bootstrap technique randomly reassigns the observations and re-computes the estimates. This process is repeated numerous times, and each iteration is treated as a repeated experiment [35].

To reduce the number of candidate terms, all non-linear and cross terms involving the residuals are omitted from the set of candidate parameters [35]. As a result, the process is now referred to as *sub-optimal* structure detection. Hence, if the the proposed model, (3.1), is the true model, its *sub-optimal* regressor matrix is:

$$\Psi(t) = \begin{bmatrix} 1 \\ V(t) \\ V^2(t) \\ V(t-1) \\ V^2(t-1) \\ z(t-1) \end{bmatrix}$$

Now, if a 2^{nd} order NARMAX model (with all the candidate parameters corresponding to $n_u = n_z = n_e = 1$ and $l = 2$) is used to estimate the parameters, the

initial sub-optimal regressor matrix is:

$$\Psi_{full}(t) = \begin{bmatrix} 1 \\ V(t) \\ V^2(t) \\ V(t-1) \\ V(t)V(t-1) \\ V^2(t-1) \\ z(t-1) \\ V(t)z(t-1) \\ V(t-1)z(t-1) \\ z^2(t-1) \end{bmatrix}$$

For a 5th order NARMAX model, the initial sub-optimal regressor matrix will contain a very large number of terms, which will include additional (3rd, 4th, and 5th order) terms involving the input $u(t)$, delayed input $u(t-1)$, and delayed output $z(t-1)$.

Once the sub-optimal regressor matrices are set up, the least squares technique is used to estimate an unbiased parameter set for the model structure [39].

To obtain parameter statistics, namely to determine which parameters are significant and which ones are spurious, the Bootstrap technique is used. A parameter is deemed spurious if 0 lies in its 99% confidence interval. The goal of this task is to study the statistics of all candidate parameters and then eliminate those that are not necessary for describing the input-output relationship of the model.

For a measurement set of 40000 samples (from the artificial system described in

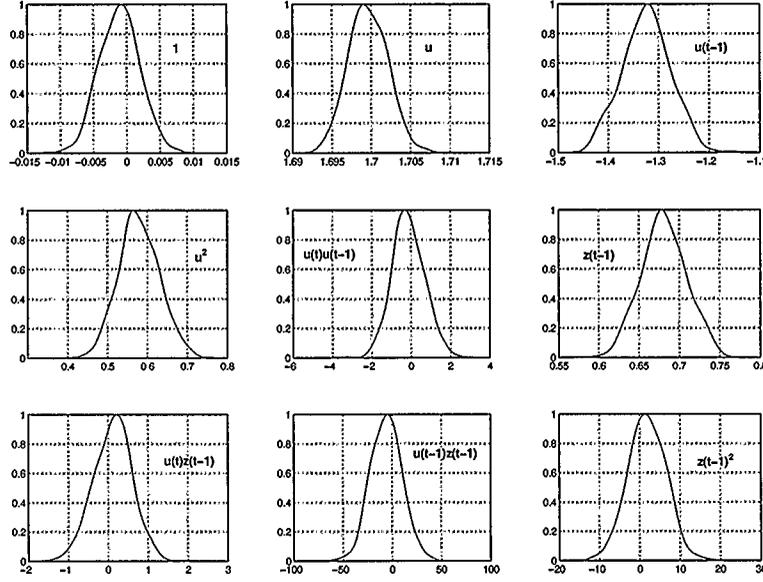


Figure 3.7: Parameter distributions for the 2^{nd} order NARMAX model

Section 3.3.2), the sub-optimal Bootstrap structure detection technique is applied to the 2^{nd} and 5^{th} order NARMAX models. 500 Bootstrap iterations are used to determine the parameter estimates. Figures 3.7 and 3.8 show the parameter distributions for the 2^{nd} and 5^{th} order NARMAX models, respectively.

As shown in Figure 3.7, for cross-product terms ($u(t)u(t-1)$, $u(t)z(t-1)$, $u(t-1)z(t-1)$) and non-linear delayed output term ($z^2(t-1)$), the parameter estimates can not be distinguished from zero. (i.e 0 lies in the 99% confidence interval of the estimates). Therefore, the parameters corresponding to these terms can be considered to be spurious, which is in agreement with the proposed model. In other words, eliminating the $u(t)u(t-1)$, $u(t)z(t-1)$, $u(t-1)z(t-1)$, and $z^2(t-1)$ terms from $\Psi_{full}(t)$ produces the proposed regressor matrix, $\Psi(t)$. The parameters correspond-

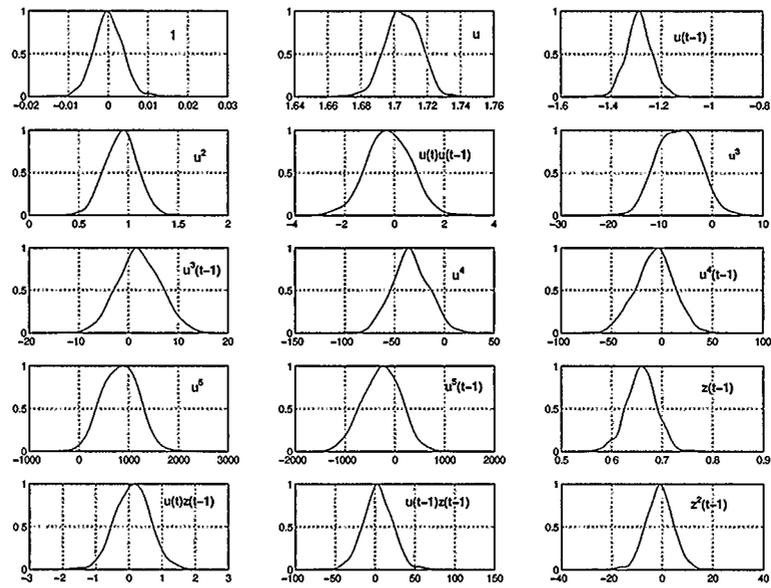


Figure 3.8: Parameter distributions for the 5th order NARMAX model

ing to the aforementioned terms display the same characteristics in the 5th order NARMAX model, shown in Figure 3.8.

It should also be noted that the DC term can not be distinguished from zero since the measurements do not contain any DC value. (The mean of input and output sequences are subtracted from them before the structure detection scheme).

The results in Figure 3.8 also indicate that a higher order polynomial is not required to approximate the exponential terms in the model. It can be seen that 0 lies in the 99% confidence interval for the parameters corresponding to $u^3(t)$, $u^3(t-1)$, $u^4(t)$, $u^4(t-1)$, $u^5(t)$, and $u^5(t-1)$. Therefore, those terms can also be eliminated from the regressor matrix as well.

3.3.5 Field Data Results

To further test the proposed technique, field data from a Swedish paper mill were applied to the model. The load voltage was varied by the mill generators in a smooth manner through a $\pm 3\%$ range while load voltage and current were measured. Active and reactive power demand were calculated off-line [33].

The cost function for the field data with one of the parameters fixed ($\theta_2 = 2.345$) is shown in Figure 3.9. It can be seen that the solution space for this problem is non-convex and non-linear. As a result, optimization routines may converge to sub-optimal local minima, resulting in inaccurate parameter estimates.

The results of Figure 3.9 emphasize the importance of the multi-stage approach since the initial estimates from the NARMAX model (stages 1 and 2) allowed the Levenberg-Marquardt algorithm to find the optimal solution. The results were verified by an extensive search of the solution space. It should be highlighted that some

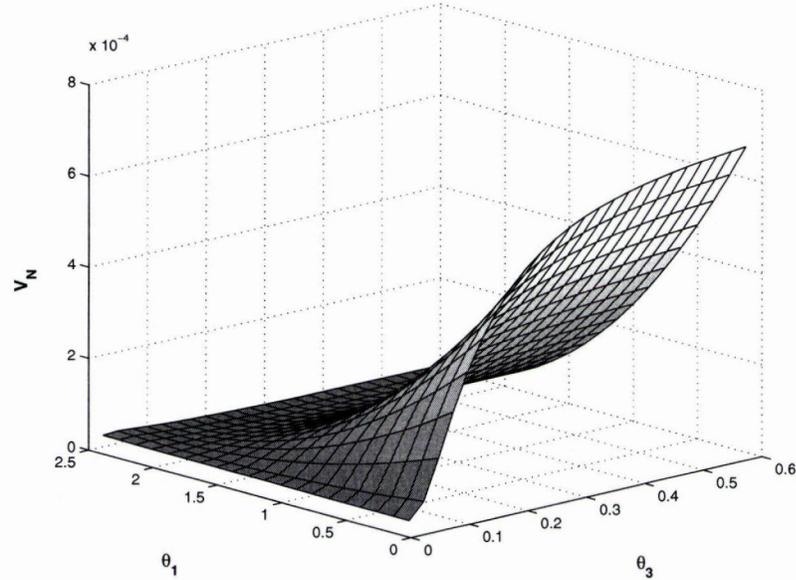


Figure 3.9: Error surface for field data with θ_2 fixed at 2.345. θ_1 and θ_3 are the varying parameters, and V_N is normalized 2-norm of the prediction errors

approaches may fail to find this solution because of poor starting points. For example, the approach presented in [33] may become trapped in a local solution depending on the initial condition. Although, theoretically, given infinite time, it should always find the global optimum.

Once the parameters of the real system are estimated, the predicted output is constructed using the dynamic load model. The residuals, ϵ , have a mean of 2.7×10^{-3} and a variance of 1.998×10^{-5} .

3.4 Ankle Dynamics Model Identification

In this section, multi-stage system identification is used, again, to overcome difficulties associated with non-linear and non-convex problems. The output of the

discrete-time version of the ankle dynamics model (shown in Figure 2.7) is a non-linear function of the model parameters (I , B , K , g , ζ , and ω) and the resulting optimization is non-convex. Therefore, a multi-stage approach, similar to that of Section 3.3, is proposed, where the initial stages are utilized to determine starting values for the non-linear optimization routine which determines the optimal physical parameters.

3.4.1 Proposed Multi-stage Identification Algorithm

Using the proposed multi-stage algorithm of Section 3.2, the main steps for identifying the ankle dynamics model are:

1. The ankle model from Section 2.6.2 is converted to a NARMAX model.
2. The NARMAX formulation, along with the input and output measurements, are used to determine the NARMAX parameters, $b_0 - b_{11}$, and the discrete delay value, τ .
3. The NARMAX parameters are mapped to estimated physical (actual) parameters of the ankle dynamics model.
4. The estimated physical parameters are used as initial values in a local optimization routine (Levenberg-Marquardt algorithm).

The remainder of this section gives details of all the stages in the proposed algorithm. Stage 1 of the proposed multi-stage algorithm is based on the NARMAX formulation of the ankle model presented in [36].

Stage 1: NARMAX formulation of the ankle model [36]

To derive the NARMAX formulation of the model, it needs to be recast in discrete-time. Two different tools are used for the conversion: Newton's Backward Formula and the Bilinear Transform.

Newton's Backward Formula, which is the discrete-time approximation to a derivative, is defined as [6]:

$$s = \frac{d u(t)}{d(t)} \approx \frac{u(kT) - u(kT - T)}{T} \quad (3.11)$$

where T is the sampling rate and k is the data point index.

The Bilinear Transform is defined as [4, 27]:

$$s = \frac{2}{T} \left(\frac{z - 1}{z + 1} \right) \quad (3.12)$$

Newton's Backward Formula is used to convert the first derivative in the linear path of the model. This provides a reasonable approximation to the continuous-time derivative given that the signal is limited to low frequencies. For this identification process, the input signal is, in fact, band-limited to lower frequencies.

The muscle activation filter in the non-linear path ($\frac{g\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$) is converted to discrete-time using the Bilinear Transform. The second order all-zero system representing intrinsic stiffness in the linear path ($I s^2 + B s + K$) is converted by using Newton's Backward since the Bilinear Transform would result in an unstable discrete equivalent.

As well, the following steps are taken to complete the transformation of the continuous-time model of Figure 2.7 to a NARMAX model:

i) The continuous-time delay is converted to discrete-time as:

$$\tau = \left\lceil \frac{\Delta}{T} \right\rceil \quad (3.13)$$

where Δ is the continuous-time delay and T is the sampling period.

ii) The half-wave rectifier in the non-linear path of the model is approximated with a second-order polynomial:

$$c_0 + c_1x(kT) + c_2x^2(kT)$$

using a least square fit.

iii) Measurement and modeling noise is represented as output additive noise:

$$z(kT) = y(kT) + e(kT)$$

Therefore, the resulting NARMAX representation of the ankle dynamics is obtained (for the remainder of this section, t is implied to be at discrete-time intervals):

$$\begin{aligned} z(t) = & b_0 + b_1z(t-1) + b_2z(t-2) + b_3u(t) + b_4u(t-1) + b_5u(t-2) + b_6u(t-3) + \\ & b_7u(t-4) + b_8v_1(t) + b_9v_2(t) + b_{10}e(t-1) + b_{11}e(t-2) \end{aligned} \quad (3.14)$$

where the regressors $v_1(t)$ and $v_2(t)$ are defined as:

$$v_1(t) = u(n - \tau) + u(n - \tau - 1) - u(n - \tau - 2) - u(n - \tau - 3) \quad (3.15)$$

$$\begin{aligned} v_2(t) &= u^2(n - \tau) + 3u^2(n - \tau - 1) + 3u(n - \tau - 2) + u^2(n - \tau - 3) - \\ &2u(n - \tau)u(n - \tau - 1) - 4u(n - \tau - 1)u(n - \tau - 2) - \\ &2u(n - \tau - 2)u(n - \tau - 3) \end{aligned} \quad (3.16)$$

The relationship between the discrete-time NARMAX coefficients $b_0 - b_{11}$ and the continuous-time parameters is as follows [36]:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \\ b_9 \\ b_{10} \\ b_{11} \end{bmatrix} = \begin{bmatrix} \frac{4c_0g\omega^2T^2}{4+\omega^2T^2+4\zeta\omega T} \\ \frac{8-2\omega^2T^2}{4+\omega^2T^2+4\zeta\omega T} \\ \frac{4\zeta\omega T-4-\omega^2T^2}{4+\omega^2T^2+4\zeta\omega T} \\ \frac{I}{T^2} + \frac{B}{T} + K \\ \frac{-2I}{T^2} - \frac{B}{T} - \left(\frac{8-2\omega^2T^2}{4+\omega^2T^2+4\zeta\omega T}\right)\left(\frac{I}{T^2} + \frac{B}{T} + K\right) \\ \frac{I}{T^2} - \left(\frac{8-2\omega^2T^2}{4+\omega^2T^2+4\zeta\omega T}\right)\left(\frac{-2I}{T^2} - \frac{B}{T}\right) - \\ \left(\frac{4\zeta\omega T-4-\omega^2T^2}{4+\omega^2T^2+4\zeta\omega T}\right)\left(\frac{I}{T^2} + \frac{B}{T} + K\right) \\ -\left(\frac{8-2\omega^2T^2}{4+\omega^2T^2+4\zeta\omega T}\right)\left(\frac{I}{T^2}\right) - \left(\frac{4\zeta\omega T-4-\omega^2T^2}{4+\omega^2T^2+4\zeta\omega T}\right)\left(\frac{-2I}{T^2} - \frac{B}{T}\right) \\ -\frac{4\zeta\omega T-4-\omega^2T^2}{4+\omega^2T^2+4\zeta\omega T}\left(\frac{I}{T^2}\right) \\ \frac{c_1g\omega^2T^2}{(4+\omega^2T^2+4\zeta\omega T)T} \\ \frac{c_2g\omega^2T^2}{(4+\omega^2T^2+4\zeta\omega T)T^2} \\ \frac{8-2\omega^2T^2}{4+\omega^2T^2+4\zeta\omega T} \\ \frac{4\zeta\omega T-4-\omega^2T^2}{4+\omega^2T^2+4\zeta\omega T} \end{bmatrix} \quad (3.17)$$

Stage 2: NARMAX parameter estimation

The input-output relationship of the model, as defined in (3.14), can be re-written as:

$$z(t) = \mathbf{x}(t, \boldsymbol{\phi})^T \boldsymbol{\phi} + e(t) \quad (3.18)$$

where

$$\mathbf{x}(t, \boldsymbol{\phi}) = \begin{bmatrix} 1 \\ z(t-1) \\ z(t-2) \\ u(t) \\ u(t-1) \\ u(t-2) \\ u(t-3) \\ u(t-4) \\ v_1(t, \tau) \\ v_2(t, \tau) \\ \epsilon(t-1, \boldsymbol{\phi}) \\ \epsilon(t-2, \boldsymbol{\phi}) \end{bmatrix} \quad \boldsymbol{\phi} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{10} \\ b_{11} \end{bmatrix} \quad (3.19)$$

where $v_1(t, \tau)$ and $v_2(t, \tau)$ are defined in (3.15), and the error terms $e(t)$ in the regressor vector are replaced with residual terms $\epsilon(t)$. Also, for a *set* of n measurements, the regressor matrix, \mathbf{X} , is defined as in (3.4).

Once again, an Extended Least Squares method is used since the regressor matrix depends on the model parameters. Also, since the regressor matrix depends on the discrete delay τ , which is unknown at this stage, the ELS procedure needs to be repeated with a number of different values of τ in order to determine the optimal delay value. Once the ELS process is performed with all possible candidate values for τ , the trial with the lowest estimation error will indicate the optimal delay value.

Stage 3: Mapping between NARMAX coefficients and physical parameters

With the value of τ determined from the previous stage, the input signals to the half-wave rectifier block, $V(t - \tau)$, can be determined by differentiating and delaying the input signal $u(t)$. Consequently, the output of the rectifier block, $\chi(t)$ can be determined as well. Once $V(t - \tau)$ and $\chi(t)$ are calculated, a non-linear least squares curve-fitting algorithm can be used to estimate c_0 , c_1 , and c_2 , the polynomial coefficients used in approximating the half-wave rectifier with a 2^{nd} order polynomial in the NARMAX formulation.

With $[\hat{b}_0 \cdots \hat{b}_{11}]$ obtained using ELS, the relationships defined in (3.17) can be used to map the NARMAX parameters to physical parameters. This process is as follows:

1. The following set of equations, originating from the 2^{nd} and 3^{rd} elements of (3.17), is solved using a non-linear least squares algorithm to determine the estimates for ω and ζ (ie. $\hat{\omega}$ and $\hat{\zeta}$ respectively):

$$0 = \hat{b}_1 - \frac{8 - 2\hat{\omega}^2 T^2}{4 + \hat{\omega}^2 T^2 + 4\hat{\zeta}\hat{\omega}T}$$

$$0 = \hat{b}_2 - \frac{4\hat{\zeta}\hat{\omega}T - 4 - \hat{\omega}^2 T^2}{4 + \hat{\omega}^2 T^2 + 4\hat{\zeta}\hat{\omega}T}$$

2. An estimate for I is evaluated by manipulating 3rd and 8th elements of (3.17) to obtain:

$$\hat{I} = -T^2 \cdot \frac{\hat{b}_7}{\hat{b}_2}$$

3. With $\hat{\omega}$, $\hat{\zeta}$, and \hat{I} determined in steps 1 and 2, the following set of equations – from the 4th and 5th elements of (3.17) – is solved using a non-linear least squares algorithm to determine the estimates for B and K (ie. \hat{B} and \hat{K} respectively):

$$0 = \hat{b}_3 - \frac{\hat{I}}{T^2} + \frac{\hat{B}}{T} + \hat{K}$$

$$0 = \hat{b}_4 - \frac{-2\hat{I}}{T^2} - \frac{\hat{B}}{T} - \left(\frac{8 - 2\hat{\omega}^2 T^2}{4 + \hat{\omega}^2 T^2 + 4\hat{\zeta}\hat{\omega}T} \right) \left(\frac{\hat{I}}{T^2} + \frac{\hat{B}}{T} + \hat{K} \right)$$

4. Finally, \hat{g} can be calculated from \hat{b}_0 since all the other values for the 1st element in (3.17) have been determined.

Stage 4: Levenberg-Marquardt Algorithm

In the final stage, the estimates of the physical parameters $\hat{\theta} = [\hat{I} \ \hat{B} \ \hat{K} \ \hat{g} \ \hat{\omega} \ \hat{\zeta}]$, found via Stage 3, are used as initial values for a Levenberg-Marquardt routine that determines their optimal values.

Figure 3.10 shows the discrete-time version of the ankle dynamics model that is used for the Levenberg-Marquardt algorithm. The input-output relationship for this model is:

$$z(t) = \left(\left(\frac{I}{T^2} + \frac{B}{T} + K \right) - \left(\frac{2I}{T^2} + \frac{B}{T} \right) q^{-1} + \frac{I}{T^2} q^{-2} \right) u(t) + \left(\frac{d_3(1 + 2q^{-1} + q^{-2})}{1 + d_1q^{-1} + d_2q^{-2}} \right) \chi(t) + e(t) \quad (3.20)$$

where the filter coefficients in the non-linear path (d_1 , d_2 , and d_3) are defined as:

$$\begin{aligned} d_1 &= \frac{2\omega^2 T^2 - 8}{4 + 4\zeta\omega T + \omega^2 T^2} \\ d_2 &= \frac{4 - 4\zeta\omega T + \omega^2 T^2}{4 + 4\zeta\omega T + \omega^2 T^2} \\ d_3 &= \frac{g\omega^2 T^2}{4 + 4\zeta\omega T + \omega^2 T^2} \end{aligned} \quad (3.21)$$

The components of each block in Figure 3.10 are obtained by using the steps detailed in Stage 1 of Section 3.4.1. The input to the non-linear path, $\chi(t)$ is also available once Stage 1 of the algorithm is completed. More specifically, once the value of the discrete delay τ is determined, the measured input $u(t)$ can be manipulated

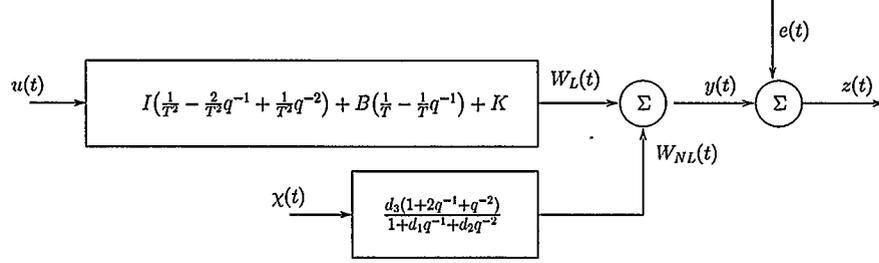


Figure 3.10: Discrete-Time model of ankle dynamics

to obtain $\chi(t)$.

To perform the optimization routine, the equations for the Jacobian need to be derived. One row of the Jacobian for this problem is defined as:

$$J(t, \cdot) = \left[\frac{\partial \hat{z}(t)}{\partial I} \quad \frac{\partial \hat{z}(t)}{\partial B} \quad \frac{\partial \hat{z}(t)}{\partial K} \quad \frac{\partial \hat{z}(t)}{\partial g} \quad \frac{\partial \hat{z}(t)}{\partial \omega} \quad \frac{\partial \hat{z}(t)}{\partial \zeta} \right]$$

These terms are determined by using $\hat{z}(t) = z(t) - e(t)$ and the relationship in (3.20):

$$\begin{aligned} \frac{\partial \hat{z}(t)}{\partial I} &= \frac{1}{T^2} (1 - 2q^{-1} + q^{-2}) u(t) \\ \frac{\partial \hat{z}(t)}{\partial B} &= \frac{1}{T} (1 - q^{-1}) u(t) \\ \frac{\partial \hat{z}(t)}{\partial K} &= u(t) \\ \frac{\partial \hat{z}(t)}{\partial g} &= \left(\frac{\omega^2 T^2}{4 + 4\zeta\omega T + \omega^2 T^2} \right) \frac{1 + 2q^{-1} + q^{-2}}{1 + d_1 q^{-1} + d_2 q^{-2}} \chi(t) \\ \frac{\partial \hat{z}(t)}{\partial \omega} &= \frac{4g\omega T^2 \left((2 + \zeta\omega T)q^2 - 4q + (2 - \zeta\omega T) \right) (q^2 + 2q + 1)}{\left((4 + 4\zeta\omega T + \omega^2 T^2)q^2 + (2\omega^2 T^2 - 8)q + (4 - 4\zeta\omega T + \omega^2 T^2) \right)^2} \chi(t) \\ \frac{\partial \hat{z}(t)}{\partial \zeta} &= \frac{-4gT^3 \omega^3 (q^4 + 2q^3 - 2q - 1)}{\left((4 + 4\zeta\omega T + \omega^2 T^2)q^2 + (2\omega^2 T^2 - 8)q + (4 - 4\zeta\omega T + \omega^2 T^2) \right)^2} \chi(t) \end{aligned}$$

Parameter	Description	Value	unit
I	Inertia	0.015	Nm/s ² /rad
B	Viscosity	0.800	Nm/rad/s
K	Elasticity	150	Nm/rad
g	Reflex Stiffness Gain	10	Nm/rad/s
ω	Natural Frequency	40.0	
ζ	Damping Parameter	1.00	

Table 3.2: Simulated ankle parameters

3.4.2 Simulated System Results

To examine the validity of the proposed algorithm, an artificial system is set up and its response is simulated. Then, the parameters of the artificial system are estimated using the multi-stage algorithm, and the predicted output is compared with the simulated output.

The input sequence for the simulated system (ankle angle) is uniformly distributed, white, zero-mean, and band-limited to the operating range of ± 0.40 radians. It is then low-pass filtered before being applied to the system, the parameters of which are shown in Table 3.2. Finally, white Gaussian noise is added to the output to account for measurement and modeling errors.

In the first experiment, a system with $N = 4000$ samples is simulated, and the sampling period is set to be 0.005 seconds. Then, the parameters of the simulated model are estimated using the proposed algorithm, and the predicted output is created accordingly. Figure 3.11 shows the simulated and predicted output for a 1 second interval of the experiment. To quantify the prediction accuracy, the percent variance accounted for (% VAF) is calculated according to [60]:

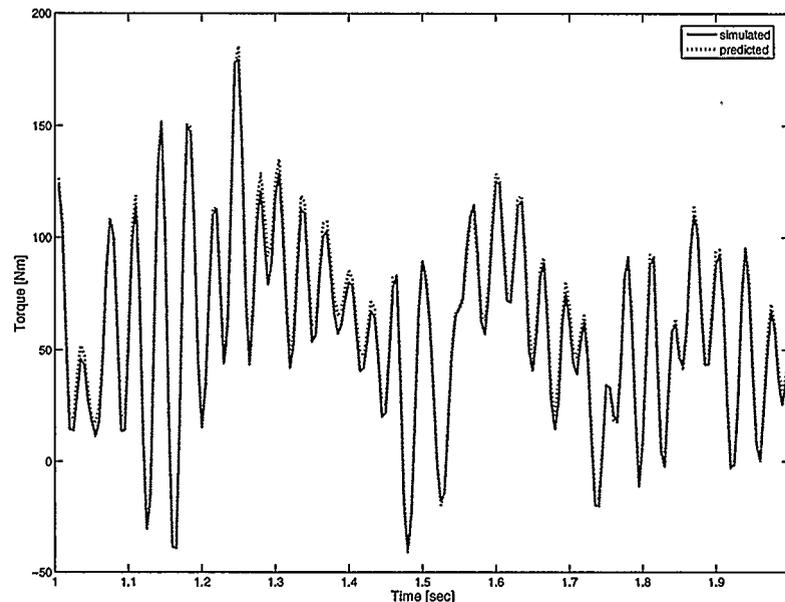


Figure 3.11: Simulated and predicted output values for the dynamic ankle model

$$\%VAF = \left(1 - \frac{\text{var}(z - \hat{z})}{\text{var}(z)} \right) \times 100$$

With a VAF of over 99%, the predicted output matches the simulated values very closely.

In the second experiment, a cross-validation technique is used to test the proposed algorithm. Five different sets of simulated measurements are created using the parameters of Table 3.2, where each set contains $N = 2000$ samples. Then, 4 of the 5 sets – collectively referred to as the Identification Set – are used to identify the model parameters by utilizing the proposed algorithm. These estimated parameters are then used to construct the predicted output for the remaining set – referred to

Identification Set	2, 3, 4, 5	1, 3, 4, 5	1, 2, 4, 5	1, 2, 3, 5	1, 2, 3, 4
Validation Set	1	2	3	4	5
I	0.0150	0.0150	0.0150	0.0150	0.0150
B	0.8006	0.7993	0.8058	0.7983	0.7994
K	149.6455	149.4522	149.2154	149.8292	149.4456
g	10.2871	10.3488	9.8537	10.3547	10.4648
ω	37.0749	36.8096	43.3814	36.6511	35.9920
ζ	0.9643	0.9688	1.0681	0.9632	0.9605
$\frac{1}{N} \sum \epsilon_{id}^2$	9.0157	11.3722	9.3039	12.1113	13.9574
$\frac{1}{N} \sum \epsilon_{val}^2$	8.7246	12.0398	10.5766	11.8180	14.3728

Table 3.3: Estimated parameters and residuals obtained from the Cross-Validation experiment

as the Validation Set. The residuals are calculated for both data sets: ϵ_{id} for the identification set, and ϵ_{va} for the validation set. The results of the cross-validation experiment are shown in Table 3.3. Since the error for the two sets are very similar, it can be concluded that the algorithm was successful in predicting the output of the validation set.

In the final experiment, Monte Carlo simulation is used to test the proposed algorithm. In this experiment, a single set of artificial input/output data is created. Then, during each run of the Monte Carlo simulation, the output set is contaminated with white Gaussian noise to obtain an SNR of 30 dB. Table 3.4 demonstrates the parameter statistics when 1000 Monte Carlo runs are used on an artificial data set containing 4000 points. The parameters obtained from the proposed algorithm have an error of less than 6%, and the theoretical values fall within two standard deviations of the estimated parameters.

Parameter	I	B	K	g	ω	ζ
Mean	0.0150	0.7937	150.0190	10.5329	37.7334	0.9642
standard deviation	0.0000	0.0034	0.3970	0.2054	1.2755	0.0046

Table 3.4: Ankle model parameters from 1000 Monte Carlo trials with SNR = 30 dB

3.5 Summary

A multi-stage identification algorithm is proposed in this chapter. The algorithm is then utilized to identify the parameters for models of non-linear dynamic power system loads and human ankle dynamics. The proposed algorithm is composed of two main blocks. In the first block, initial estimates of the actual parameters are obtained by fitting a simplified model. In the second block, these initial estimates are used in a Levenberg-Marquardt optimization approach to find the parameter values for the non-linear system.

The proposed technique was significantly faster than the ASA algorithm in estimating the load model parameters. In both simulated experiments and experiments based on actual load data, the proposed technique found the global optimum. The proposed technique converged to the CRLB showing that it can consistently find the optimal parameters given sufficient data. The choice of NARMAX model for the power stem loads was examined by using polynomial verification and Bootstrap structure detection. The proposed algorithm was also successful in creating accurate estimates of the ankle dynamics.

Chapter 4

System Identification with Variable Decomposition

4.1 Introduction

As presented in Chapter 2, the main goal of most identification problems is to find the optimal parameter set, θ^* such that:

$$\theta^* = \arg \min_{\theta} V_N(\theta) \quad (4.1)$$

where N is the number of data samples (measurements). The cost function, $V_N(\cdot)$ is given by:

$$V_N(\theta) = \frac{1}{2} \|\epsilon(\theta)\|_2^2 \quad (4.2)$$

where $\|x\|_2$ is the 2-norm of the vector x and ϵ is the difference between the measured output z and the predicted output \hat{z} , i.e. $\epsilon(\theta) = z - \hat{z}(\theta)$.

Considering the following problem:

$$\min_{\theta \in \mathbb{R}^m} V_N(\theta) \quad (4.3)$$

where m is the number of parameters to be identified, if one can partition θ into two sets:

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_c \\ \theta_{nc} \end{pmatrix} \quad (4.4)$$

where the identification problem is convex in $\theta_c \in \mathbf{R}^p$ (for any valid choice of θ_{nc}) and non-convex in $\theta_{nc} \in \mathbf{R}^q$ ($p + q = m$), then the following sub-problem:

$$\theta_c(\theta_{nc}) = \arg \min_{\alpha \in \mathbf{R}^p} V_N(\alpha, \theta_{nc}) \quad (4.5)$$

is easy to solve for every fixed θ_{nc} in the domain. One can replace the original m -dimensional problem, (4.3), with the following q -dimensional problem:

$$\min_{\theta_{nc} \in \mathbf{R}^q} \widetilde{V}_N(\theta_{nc}) \quad (4.6)$$

where

$$\widetilde{V}_N(\theta_{nc}) = V_N(\theta_c(\theta_{nc}), \theta_{nc}) \quad (4.7)$$

and $\theta_c(\theta_{nc})$ is the solution of (4.5) [21, 49, 56].

Considering the non-linear load model described in Chapter 2:

$$\hat{z}(k\ell) = P_o \frac{1 - e^{-\theta_3 \ell}}{q - e^{-\theta_3 \ell}} (V^{\theta_1}(k\ell) - V^{\theta_2}(k\ell)) + P_o V^{\theta_2}(k\ell) \quad (4.8)$$

if the problem is shown to be convex with respect to θ_1 and θ_2 , the 3-dimensional problem:

$$\min_{\boldsymbol{\theta} \in \mathbf{R}^3} V_N(\boldsymbol{\theta}) \quad (4.9)$$

where $\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \theta_3]$, can be simplified into a 1-dimensional problem:

$$\min_{\theta_3 \in \mathbf{R}^1} \widetilde{V}_N(\theta_3) \quad (4.10)$$

where

$$\widetilde{V}_N(\theta_3) = V_N(\theta_{1,2}(\theta_3), \theta_3) \quad (4.11)$$

However, each evaluation of (4.10) needs the solution of the following 2-dimensional (convex) problem:

$$\theta_{1,2}(\theta_3) = \arg \min_{\alpha \in \mathbf{R}^2} V_N(\alpha, \theta_3) \quad (4.12)$$

In summary, if the problem is shown to be convex with respect to θ_1 and θ_2 , then the solution space can effectively be considered a function of θ_3 only. In other words, if $V_N(\theta_1, \theta_2)$ is convex for a fixed θ_3 , then $V_N(\boldsymbol{\theta})$ of (4.2) can be transformed into $\widetilde{V}_N(\theta_3)$. As a result, a one dimensional search in θ_3 can be used to determine the optimal parameter set.

The next step is to demonstrate the convexity of the solution space with respect to θ_1 and θ_2 .

4.2 Convexity

Two different approaches in establishing the convexity of the solution space with respect to θ_1 and θ_2 are presented in this section:

1. showing that the Hessian is positive definite, and

2. using an exponential expansion method.

4.2.1 Hessian

One approach to show that the solution space is convex in θ_1 and θ_2 is to show that the Hessian of (4.5) is positive definite everywhere on the domain [10]. For this sub-problem, the Hessian is defined as:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 V_N}{\partial \theta_1^2} & \frac{\partial^2 V_N}{\partial \theta_1 \partial \theta_2} \\ \frac{\partial^2 V_N}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 V_N}{\partial \theta_2^2} \end{bmatrix} \quad (4.13)$$

The cost function, as defined in (4.2), is:

$$\begin{aligned} V_N &= \frac{1}{2N} \sum_{t=1}^N \epsilon(\boldsymbol{\theta}, t)^2 \\ &= \frac{1}{2N} \sum_{t=1}^N (z(t) - \hat{z}(\boldsymbol{\theta}, t))^2 \end{aligned} \quad (4.14)$$

For the remainder of this section, the explicit time dependence is removed to clarify the presentation. For example, the summations presented in the following equations are over the record length (time). The elements of the Hessian (4.13), therefore, are:

$$\begin{aligned}
\frac{\partial^2 V_N}{\partial \theta_1^2} &= \frac{1}{N} \sum^N F(\theta_3)^2 P_o^2 (V^{\theta_1})^2 (\log V)^2 - \frac{1}{N} \sum^N F(\theta_3) P_o V^{\theta_1} (\log V)^2 \cdot \\
&\quad (z - F(\theta_3) P_o (V^{\theta_1} - V^{\theta_2}) + P_o V^{\theta_2}) \\
\frac{\partial^2 V_N}{\partial \theta_2^2} &= \frac{1}{N} \sum^N (F(\theta_3) P_o (V^{\theta_2}) \log V - P_o V^{\theta_2} \log V)^2 + \frac{1}{N} \sum^N \left(F(\theta_3) P_o (V^{\theta_2}) (\log V)^2 \right. \\
&\quad \left. - (z - F(\theta_3) P_o (V^{\theta_1} - V^{\theta_2}) + P_o V^{\theta_2}) \cdot P_o V^{\theta_2} (\log V)^2 \right) \\
\frac{\partial^2 V_N}{\partial \theta_1 \partial \theta_2} &= \frac{-1}{N} \sum^N F(\theta_3) P_o V^{\theta_1} (\log V)^2 \cdot (F(\theta_3) P_o (V^{\theta_2}) - P_o V^{\theta_2})
\end{aligned}$$

where, $F(\theta_3)$ is defined as:

$$F(\theta_3) = \frac{1 - e^{-\theta_3 \ell}}{q - e^{-\theta_3 \ell}} \quad (4.15)$$

and was introduced to allow for a more compact representation.

The Hessian is positive definite (and the sub-problem is convex) if and only if all of its eigenvalues are positive [10]. Since the Hessian for this sub-problem is a 2×2 matrix (4.13), there are 2 eigenvalues to be determined. The symbolic expressions for the eigenvalues do not explicitly indicate that they are positive (since they are non-trivial). Therefore, numerical examination is utilized in order to establish convexity. More specifically, for any fixed value of θ_3 , the eigenvalues need to be determined for the entire parameter space of θ_1 and θ_2 , which is [1]:

$$\begin{aligned}
0 &\leq \theta_1 \leq 3 \\
0.5 &\leq \theta_2 \leq 2.5
\end{aligned} \quad (4.16)$$

If the eigenvalues are shown to be positive over the entire parameter range, then the Hessian for the sub-problem will be positive definite.

Figure 4.1 exhibits the eigenvalues for a sample data set: $\theta_{simulated} = [1.2 \ 2.2 \ 0.2]$, $N = 5000$, input (V) uniformly distributed between 0.9 and 1.1, and $SNR = 30 \text{ dB}$. With θ_3 fixed at the theoretical value of 0.2, θ_1 and θ_2 are changed in small increments within the permissible range and the eigenvalues are recorded at every step. It can be seen that the eigenvalues are positive over the entire range of θ_1 and θ_2 (and a fixed θ_3) as illustrated in Figure 4.1. The solution space for the aforementioned simulation, shown in Figure 4.2, is convex, as expected.

So far, the convexity of the sub-problem for one theoretical data set ($\theta_{simulated} = [1.2 \ 2.2 \ 0.2]$) has been established. To extend this conclusion to the entire parameter space (permissible range for θ), 1190 numerical simulations for examining the eigenvalues were performed for different values of θ_3 . All of these simulations provided equivalent results to those shown in Figures 4.1 and 4.2.

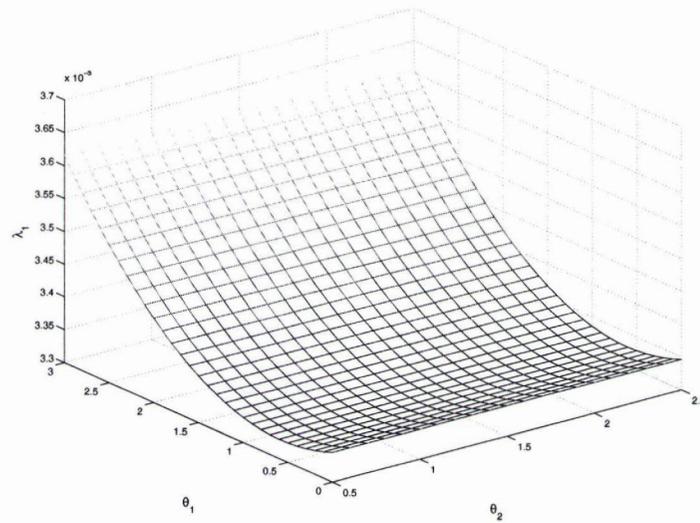
Therefore, the convexity of the solution space in $\theta_{1,2}$ is numerically established.

4.2.2 Exponential Expansion

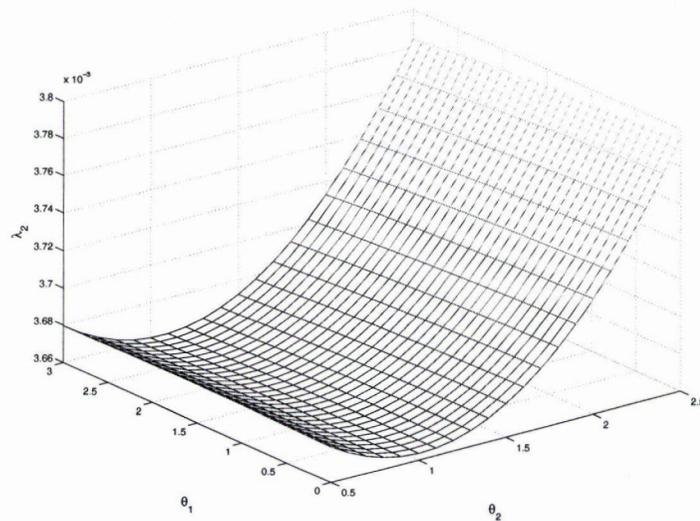
Starting from the non-linear load model described in Chapter 2 (and removing the explicit time dependence for the sake of brevity) :

$$\hat{z} = P_o \frac{1 - e^{-\theta_3 \ell}}{q - e^{-\theta_3 \ell}} (V^{\theta_1} - V^{\theta_2}) + P_o V^{\theta_2} \quad (4.17)$$

the nominal power P_0 is set to 1, and the first order filter $\frac{1 - e^{-\theta_3 \ell}}{q - e^{-\theta_3 \ell}}$ is rewritten as $\frac{(1 - e^{-\theta_3 \ell})q^{-1}}{1 - (e^{-\theta_3 \ell})q^{-1}}$. For a fixed value of θ_3 , the coefficients of the first order filter are constant.



(a) First eigenvalue



(b) Second eigenvalue

Figure 4.1: Eigenvalues, λ_1 and λ_2 , over the range of permissible θ_1 and θ_2 for $\theta_3 = 0.2$

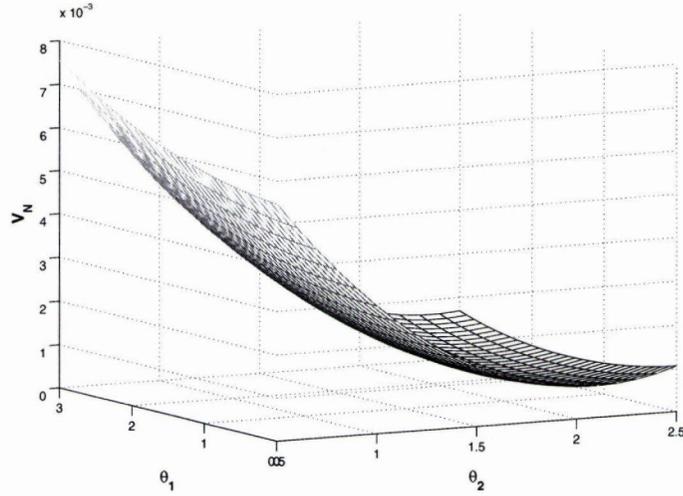


Figure 4.2: Solution space over the range of permissible θ_1 and θ_2 for $\theta_3 = 0.2$

Therefore, the load model can now be re-written as:

$$\hat{z} = \frac{aq^{-1}}{1+bq^{-1}}(V^{\theta_1} - V^{\theta_2}) + P_o V^{\theta_2} \quad (4.18)$$

where $a = 1 - e^{-\theta_3 \ell}$ and $b = -e^{-\theta_3 \ell}$ are known constants for a given θ_3 .

Re-writing the V^θ term as an exponential:

$$V^\theta = e^{\theta \log V} \quad (4.19)$$

and expanding the exponential term as a series [2] gives:

$$V^\theta = 1 + \theta \log V + \frac{1}{2}\theta^2 \log^2 V + \frac{1}{3}\theta^3 \log^3 V + \dots \quad (4.20)$$

Substituting the exponential expansion into the model of (4.18) gives:

$$\begin{aligned}
\hat{z} &= \frac{aq^{-1}}{1+bq^{-1}} \left(1 + \theta_1 \log V + \frac{1}{2}\theta_1^2 \log^2 V + \frac{1}{3}\theta_1^3 \log^3 V + \dots \right. \\
&\quad \left. - 1 - \theta_2 \log V - \frac{1}{2}\theta_2^2 \log^2 V - \frac{1}{3}\theta_2^3 \log^3 V - \dots \right) + \\
&\quad 1 + \theta_2 \log V + \frac{1}{2}\theta_2^2 \log^2 V + \frac{1}{3}\theta_2^3 \log^3 V + \dots
\end{aligned} \tag{4.21}$$

Reintroducing t to indicate time dependence and re-arranging (4.21) in a vector format gives:

$$\begin{aligned}
\hat{z}(t) &= \frac{1}{1+bq^{-1}} \mathbf{X} \Theta \\
&= \frac{1}{1+bq^{-1}} \begin{bmatrix} a \log V(t-1) \\ a \frac{1}{2} \log^2 V(t-1) \\ a \frac{1}{3} \log^3 V(t-1) \\ \vdots \\ \log V(t) - \log V(t-1) \\ \frac{1}{2}(\log V^2(t) - \log^2 V(t-1)) \\ \frac{1}{3}(\log V^3(t) - \log^3 V(t-1)) \\ \vdots \end{bmatrix}^T \begin{bmatrix} \theta_1 \\ (\theta_1)^2 \\ (\theta_1)^3 \\ \vdots \\ \theta_2 \\ (\theta_2)^2 \\ (\theta_2)^3 \\ \vdots \end{bmatrix}
\end{aligned} \tag{4.22}$$

which is a filtered linear regression, where \mathbf{X} is the regressor matrix and Θ is the parameter vector. Since the range of values for V is between 0.9 and 1.1, the higher order terms of the exponential expansion ($\frac{1}{2} \log^2 V$, $\frac{1}{3} \log^3 V$, \dots) quickly approach zero.

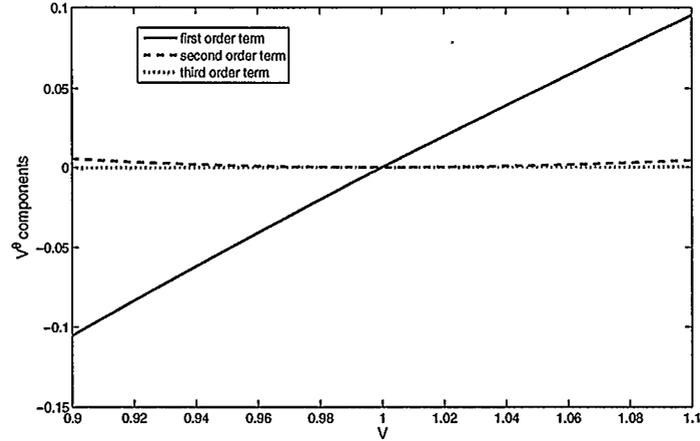


Figure 4.3: The first three terms in the regressor matrix after the exponential expansion of V^θ

Figure 4.3 shows the first three terms in the regressor matrix \mathbf{X} (regressor terms corresponding to θ_1). It can be seen that the higher order terms are relatively insignificant. The same conclusion can be drawn about the regressor terms corresponding to θ_2 . As a result, the relationship from (4.22) can be reasonably approximated as:

$$\hat{z}(t) \approx \begin{bmatrix} \frac{1}{1+bq^{-1}} (a \log V(t-1)) \\ \frac{1}{1+bq^{-1}} (\log V(t) - \log V(t-1)) \end{bmatrix}^T \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \quad (4.23)$$

which is a linear regression in θ_1 and θ_2 only. Since the linear regression is convex in its parameters (θ_1 and θ_2) [39], it can be seen that the predicted output is convex in θ_1 and θ_2 for a fixed θ_3 . Therefore, the convex optimization problem of (4.12) can be utilized.

4.3 Separation of Variables

The parameter estimation problem can be separated into two parts:

- i) The 2-dimensional and convex problem of (4.12). This problem is solved using a Levenberg-Marquardt [40] algorithm, since it avoids the problems associated with singular or poorly-conditioned Jacobians.
- ii) The 1-dimensional problem of (4.10). Two different approaches are used for this problem: a basic line search, and a “rate of change” search.

The proposed algorithm is as follows:

1. Starting from the lower end of the permissible range for θ_3 , an initial value for θ_3 is chosen:

$$\theta_3^1 = 0.01$$

where θ_3^k is the k^{th} value of θ_3 examined. (Throughout this section, superscripts indicate the iteration number for the algorithm.)

2. With θ_3 fixed at θ_3^1 , the Levenberg-Marquardt algorithm is used to determine the optimal $\theta_{1,2}$. The resulting parameter set θ_*^1 and the corresponding cost V_N^1 are recorded. (The subscript $*$ denotes the optimal parameter set, and the superscript 1 denotes the iteration number.)
3. Next, θ_3^2 is defined as:

$$\theta_3^2 = \theta_3^1 + \varepsilon \tag{4.24}$$

where ε is a user-selected step size. In this research, a step size of 0.01 is used. The same procedure as Step 2 is repeated with θ_3^2 to obtain θ_*^2 and V_N^2 .

4. Next, it is required to search the parameter space of θ_3 . Two different approaches were considered to accomplish this task: first, a simple line search approach is used, which solves for different θ_*^k and V_N^k at small intervals. Second, a “rate of change” approach is utilized, which is based on the observation that fewer points need to be considered when the objective (V_N^k) is not changing significantly with respect to θ_3 and more points should be considered when it is. These approaches are used to generate a set of solutions for the range of θ_3 . Details on the two approaches are given below:

(a) *Line Search*: θ_3 is updated according to:

$$\theta_3^k = \theta_3^{k-1} + \varepsilon$$

where ε is the user selected step size defined previously.

(b) *Rate of Change (RoC) Search*: In this approach, the slope of the objective function with respect to θ_3 is estimated using the secant from the two previous points, i.e.

$$m = \frac{\Delta V_N}{\Delta \theta_3} = \frac{V_N^{k-1} - V_N^{k-2}}{\theta_3^{k-1} - \theta_3^{k-2}} \quad (4.25)$$

where m is the approximation of the slope.

Since more points should be considered when the slope is large, the step size for θ_3 should be inversely proportional to m . Therefore, the following update for θ_3 is used:

$$\theta_3^k = \theta_3^{k-1} + K \frac{1}{m} \quad (4.26)$$

where K is a fixed scalar coefficient to control the step length. For this thesis, $K = \frac{1}{N}$, where N is the number of data samples. This approach is similar to techniques used in continuation methods to trace bifurcation diagrams [51].

For each value of θ_3 (determined by either of the two above approaches), the Levenberg-Marquardt algorithm is used to determine the optimal θ_1 and θ_2 . The resulting parameter set θ_*^k and the corresponding cost, V_N^k are recorded.

5. The parameter set θ_* that corresponds to the lowest cost is the optimal parameter set.

A flow chart, summarizing all the stages in the proposed identification process for the dynamic load model is shown in Figure 4.4

There is one potential pitfall for the RoC method: If the solution space is not very steep around the minimum point, the RoC might encounter difficulties in finding the true optimal parameter set. Figure 4.5 is used to demonstrate this problem. The RoC approach, as described here, starts from the left hand side of the solution space in Figure 4.5 and updates the θ_3 value based on the slope of the graph at each iteration. At point θ_3^k (k^{th} iteration of the technique), the slope of the graph is relatively small. Therefore, the RoC will utilize a relatively large value for the θ_3 update step, resulting in point θ_3^{k+1} . Consequently, the optimal point θ_3^* is bypassed by the algorithm.

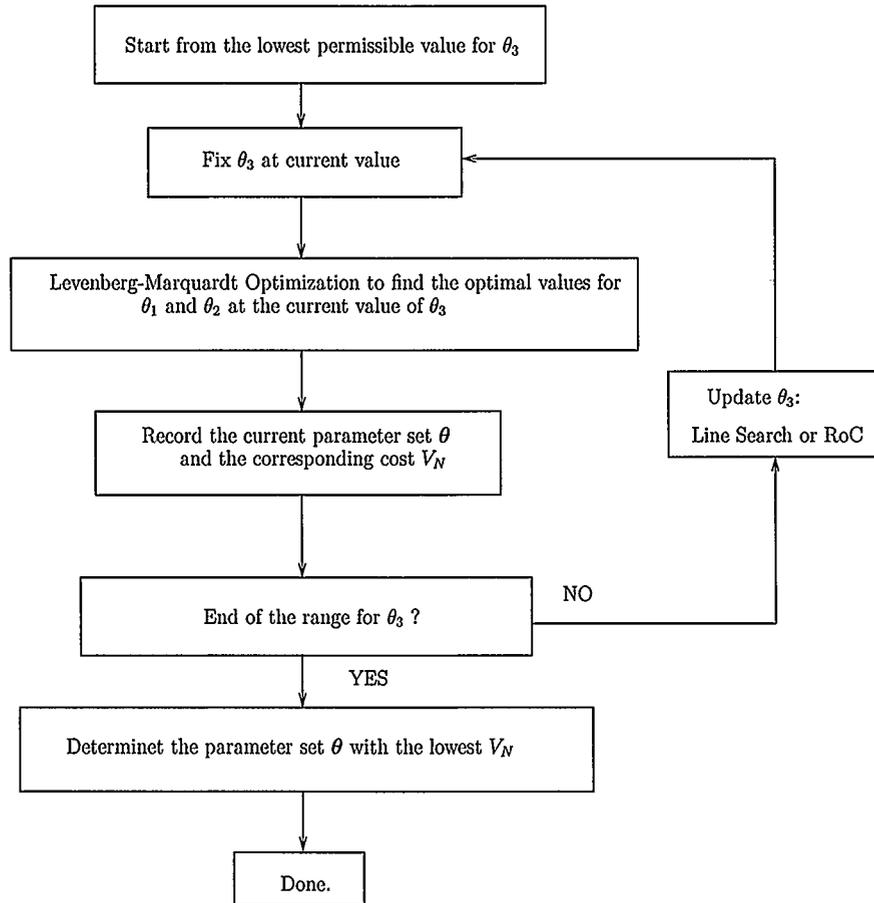


Figure 4.4: Summary of the separable identification process for the non-linear dynamic load model

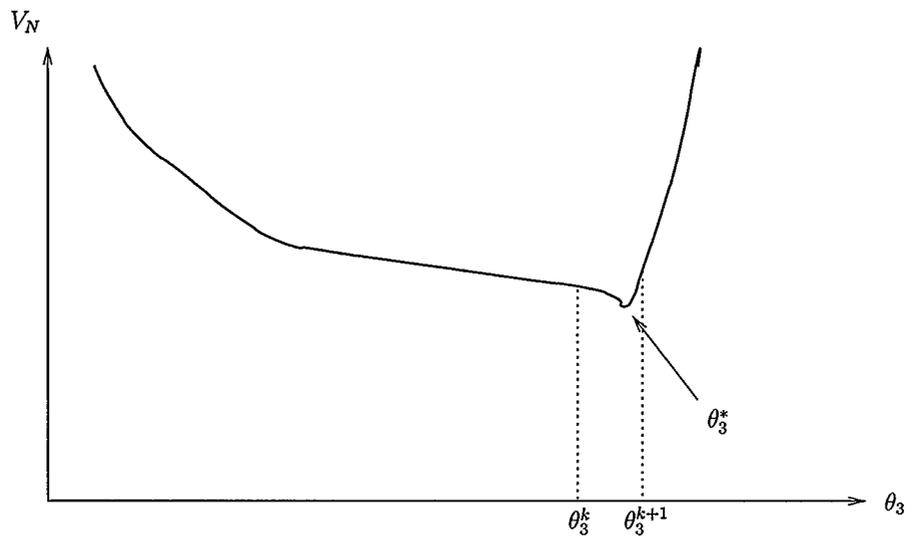


Figure 4.5: A hypothetical cost function over the range of θ_3 to demonstrate the pitfall of the RoC Method

To avoid the aforementioned problem with the RoC, a third approach, named the *Improved RoC*, is implemented. This algorithm detects the critical points from the slope of the cost function. It will then perform a line search around these critical points.

The steps for the improved RoC, are as follows:

1. The separable identification algorithm of Figure 4.4 is performed by using the RoC method to do a line search with respect to θ_3 . Since the derivative of V_N is continuous with respect to θ_3 , there must be a critical point between every pair of sequential points where the derivative changes sign.
2. The area around each of these points is examined by using a line search method to find the minima. More specifically, if the sign of the slope of V_N for the $\theta_3^{i-1} : \theta_3^i$ segment is different from that of the $\theta_3^i : \theta_3^{i+1}$ segment, then a line

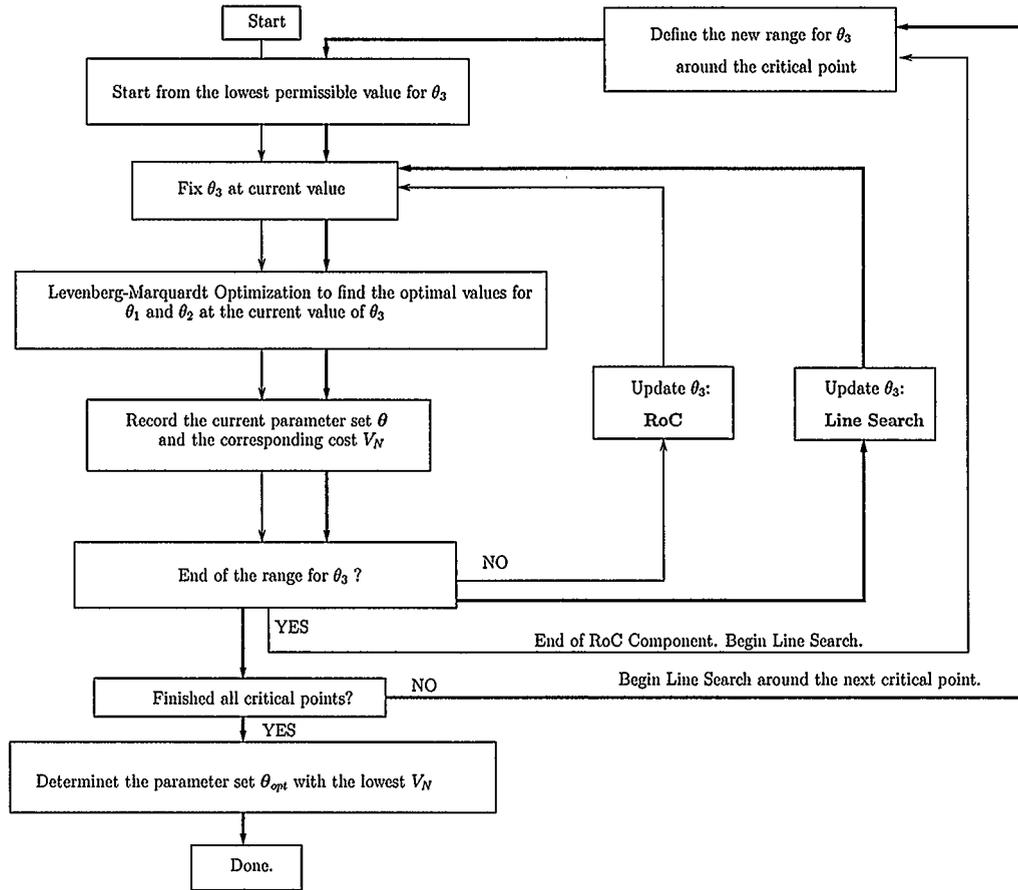


Figure 4.6: Summary of the Improved Rate of Change identification process for the non-linear dynamic model

search is performed between θ_3^{i-1} and θ_3^{i+1} .

3. The parameter set from the line search that corresponds to the lowest cost is the optimal parameter set.

A flow chart, summarizing all the stages in the Improved RoC process is shown in Figure 4.6. The bold lines in the chart indicate the new steps that are added to a normal RoC approach to obtain the Improved RoC.

A fourth approach, utilizing a Quasi-Newton Method [49] was developed. The steps for the Quasi-Newton RoC are as follows:

1. The separable identification algorithm of Figure 4.4 is performed by using the RoC method to do a line search with respect to θ_3 . Since the derivative of V_N is continuous with respect to θ_3 , there must be a critical point between every pair of sequential points where the derivative changes sign.
2. A Quasi-Newton optimization, i.e. a local optimization algorithm, is initialized at the points identified in Step 1 to locate the minima. Details on this algorithm are given in [49].

4.4 Numerical Results

In this section, the proposed algorithm is studied and its performance is investigated using simulated and real (field) data. First, the three proposed approaches for finding the optimal θ_3 – namely, the line search, the rate of change (RoC), and the Improved RoC approach – are compared with each other. Then, the performance of the proposed algorithm is compared with published approaches, namely the Adaptive Simulated Annealing technique. Next, the sensitivity of the proposed algorithm to initial estimates and actual values of the parameters is studied. Because the proposed algorithm is designed to find the global optimum, it should not be sensitive to initial or theoretical values. Finally, application of the proposed algorithm to real data is investigated.

	Actual Value	RoC Method	Improved RoC Method	Line Search Method
θ_1	2.1	2.1235	2.081	2.083
θ_2	1.1	1.0897	1.089	1.089
θ_3	0.9	0.8484	0.934	0.93
V_N		5.32×10^{-4}	5.19×10^{-4}	5.19×10^{-4}
total time		11 sec.	22 sec.	130 sec.

Table 4.1: Parameter Statistics and Time Performance of the RoC and Line Search Approaches for Finding θ_3

4.4.1 Performance Comparison

In this section, the RoC and Improved RoC approaches and the basic line search are compared with each other. The goal is to study the benefits and drawbacks of the three techniques, namely, the accuracy of the approaches versus their speed of completion. Since the basic line search method tends to examine more points than the RoC approach, it may be more accurate in finding the optimal parameters. Accordingly, the basic line search can be time-consuming. The Improved RoC provides a compromise between the other two approaches.

To perform an initial comparison, 5000 data samples were created using the theoretical parameter set $\theta = [2.1 \ 1.1 \ 0.9]$ and an SNR of 30 dB. The basic line search was performed by dividing the permissible range for θ_3 , $0 < \theta_3 \leq 5$, into intervals of 0.01 unit length and performing a Levenberg-Marquardt optimization at each point. The RoC and Improved RoC approaches were performed as described in Section 4.3. The statistics for the optimal parameters are shown in Table 4.1.

It can be seen that the RoC approach presents a major advantage with respect

to execution time, without compromising the accuracy of the estimated parameters.

Using the same set of data as those used for Table 4.1, Figures 4.7 and 4.8 show the values of θ_3 examined by each approach and their corresponding cost value. Figure 4.7 illustrates how the RoC approach considers more points when the slope is high and fewer points where the cost function does not change rapidly with respect to θ_3 (small slope). This allows for the better time performance of the RoC approach versus the line search. In the line search approach, 500 different values for θ_3 were examined since its domain was divided into uniform intervals of 0.01 unit length. However, the RoC approach only examined 43 points within the θ_3 domain since the distance between the tested points varied based on the rate of change.

Figure 4.8 shows the points examined by the RoC and Improved RoC approaches around the optimal solution. The figure illustrates how the Improved RoC approach searches the solution space around the optimal point with a higher resolution than the normal RoC method. In this example, the Improved RoC approach evaluated 43 additional points in comparison to the normal RoC.

The accuracy of the RoC technique is tied to the value used for K . For example, in Table 4.1, the value found for θ_3 was 0.8484 when $K = \frac{1}{N}$. When the RoC method was repeated with $K = \frac{1}{2N}$, the algorithm found $\theta_3 = 0.8837$ after examining 84 points, i.e. a 4 % improvement in the parameter error.

The Quasi-Newton RoC approach found the optimal parameters in two iterations, with the same accuracy as the Improved RoC technique. The execution time for the Quasi-Newton approach, 35.43 seconds, was larger than the Improved RoC due to the computations required in the step size calculations [49]. However, this characteristic can not be assumed to be true for all situations. If the desired accuracy of the

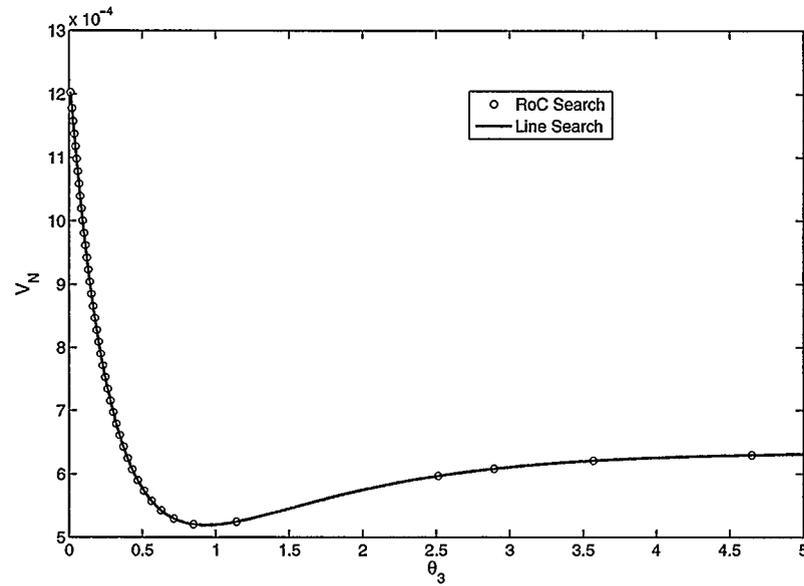


Figure 4.7: Comparison of the values of θ_3 examined by the Line search and RoC approaches

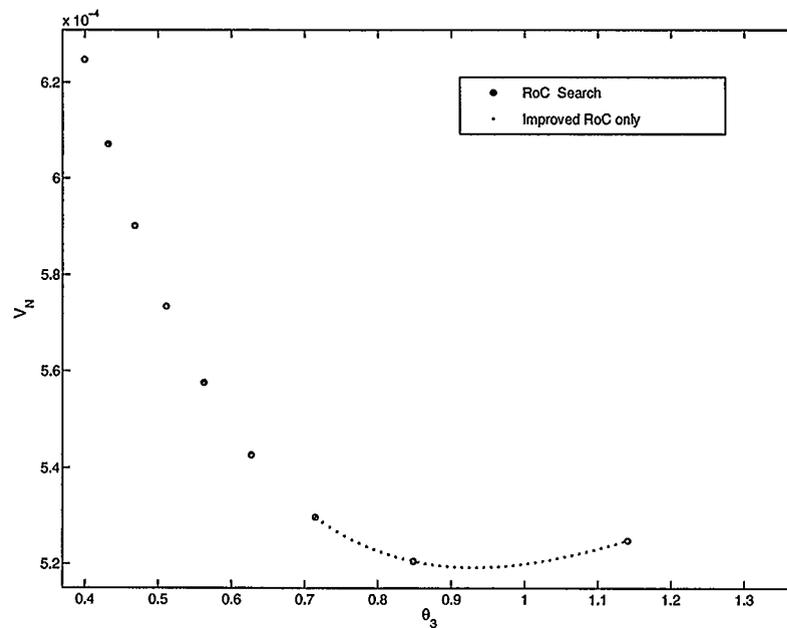


Figure 4.8: Comparison of the values of θ_3 , near the optimal solution, examined by the RoC and Improved RoC approaches.

solution was increased, the number of steps required for the Quasi-Newton method would not increase substantially, thereby improving its relative efficiency.

For the remainder of this chapter, the Improved RoC approach is used as the default approach for searching the parameter space of θ_3 . This approach was chosen for its ease of implementation and accuracy.

4.4.2 Simulated System Results

To study the proposed approach, a simulated system, identical to that of Section 3.3.2, is created. First, the results from applying the proposed approach are compared to published results. Then, a cross-validation technique is used to examine the performance of the proposed approach.

Similar to 3.3.2, the Adaptive Simulated Annealing (ASA) technique [26] is implemented for comparison purposes. The same initial parameter set as [33], [0.5 2.0 1.7], is used as the starting point for the ASA technique.

Table 4.2 demonstrates the parameter statistics when 30 Monte Carlo runs are used on an artificial data set containing 1000 points. (The number of samples and Monte Carlo runs were chosen to match those used in [33].) The sample mean and standard deviation from both approaches (ASA and separable approach) are presented in the table. It can be seen that both techniques can successfully estimate the parameters of the simulated system. The parameters obtained from ASA have an error of less than 9.8% and those from the proposed algorithm fall within 0.5% of the simulated values. Also, all simulated values are within one standard deviation of the estimated parameters.

The main difference between the two approaches is in their respective time per-

	Actual Value	Sample Mean		Sample Std Dev($\times 10^{-2}$)	
		ASA	Separable	ASA	Separable
θ_1	1.2	1.222	1.2083	7.77	4.7557
θ_2	1.7	1.7040	1.7034	2.02	1.8650
θ_3	0.4	0.3611	0.4014	6.27	7.5617

Table 4.2: Parameter statistics for ASA and the separable approach from 30 Monte Carlo runs with $N = 1000$ points.

formances. The proposed algorithm presents a major advantage over ASA in terms of computational burden. To obtain the parameters reported in Table 4.2, ASA required 1897.5 seconds while the separable identification technique required only 191.6 seconds, under identical conditions (on an AMD Athlon based PC).

To examine the proposed algorithm, a cross-validation test is performed using simulated data. The description of the cross-validation test is as follows:

Five different sets of simulated measurements are created by using theoretical parameter set $\theta = [2.1 \ 1.1 \ 0.9]$ and an SNR of 30 dB. Then, 4 of the 5 sets – collectively referred to as the Identification Set – are used to identify the model parameters by utilizing the proposed algorithm. These estimated parameters are then used to construct the predicted output for the remaining set – referred to as the Validation Set. The residuals are calculated for both data sets: ϵ_{id} for the identification set, and ϵ_{va} for the validation set. The results of the cross-validation experiment are summarized in Table 4.3.

Since the error for the two sets are very similar, it can be concluded that the algorithm was successful in predicting the output of the validation set.

Identification Set	2, 3, 4, 5	1, 3, 4, 5	1, 2, 4, 5	1, 2, 3, 5	1, 2, 3, 4
Validation Set	1	2	3	4	5
θ_1	2.0896	2.1008	2.0853	2.0865	2.0908
θ_2	1.0930	1.0912	1.0950	1.0926	1.0962
θ_3	0.9140	0.9024	0.9329	0.9085	0.9152
$\frac{1}{N} \sum \epsilon_{id}^2 (\times 10^{-3})$	1.0107	1.0094	1.0179	1.0034	1.0050
$\frac{1}{N} \sum \epsilon_{val}^2 (\times 10^{-3})$	1.0035	1.0099	0.9752	1.0345	1.0274

Table 4.3: Estimated parameters and residuals obtained from the Cross-Validation experiment for the separable identification algorithm

4.4.3 Sensitivity to Initial Values

The Levenberg-Marquardt algorithm is a modified gradient descent approach. As a result, initial parameters are generally significant in the optimization process. To study the sensitivity of the proposed algorithm, the following simulation was performed on artificial data: A set of 5000 data samples was created, and then contaminated with Gaussian noise to produce a 30 dB SNR. The proposed algorithm was used on the data set to determine the optimal parameter set. Each attempt, however, contained a different set of initial values. These values were chosen from uniform intervals inside the permissible range for θ_1 and θ_2 . In this study, 215 possible combinations for starting values were considered. All attempts, regardless of the starting points for the Levenberg-Marquardt algorithm, converged to the same optimal parameter set. This characteristic was expected since the solution space was already shown to be convex with respect to θ_1 and θ_2 .

4.4.4 Sensitivity to Theoretical Values

To examine the robustness of the proposed algorithm, a number of different theoretical parameter sets were studied. The 3-dimensional surface containing the permissible values for θ_1 , θ_2 , and θ_3 was divided into a grid with uniform intervals at 0.3 units. Then, each node from the grid was used to create a theoretical (artificial) measurement with 30 dB SNR. Each set of data was then applied to the proposed algorithm to determine the optimal parameter set. Only the Improved RoC approach was used to search the parameter space.

A total of 1190 trials were performed, and the mean, E , and standard deviation of the prediction errors are:

$$E[V_N] = 5.0859 \times 10^{-4} \quad std[V_N] = 1.18 \times 10^{-5}$$

It has been shown that the algorithm consistently found reasonable estimates for the optimal parameters.

4.4.5 Application to Field Data

To test the validity of the proposed technique, it was applied to field data from a Swedish paper mill [33]. The load voltage was varied by the mill generators in a smooth manner through a $\pm 3\%$ range while load voltage and current were measured. Active and reactive power demand were calculated off-line.

Using these measurements, the proposed algorithm was utilized to determine the physical parameters of the load. These parameters were then used to construct the estimated active power demand of the factory. The measured and predicted output

are shown in Figure 4.9.

To quantify the prediction accuracy, the percent variance accounted for (% VAF) is calculated according to [60]:

$$\% \text{VAF} = \left(1 - \frac{\text{var}(z - \hat{z})}{\text{var}(z)} \right) \times 100$$

With over 99% VAF and $V_N = 1.387 \times 10^{-5}$, the predicted output matches the field measurements with very small error.

4.5 Summary

A separable identification algorithm for nonlinear dynamic power system loads is presented in this chapter. The proposed algorithm separates the model parameters into convex and non-convex groups, thereby making the identification process significantly easier.

The algorithm is shown to accurately determine the parameters of aggregated power loads. Simulated (artificial) systems and real (field) data were investigated and the optimal parameters were found in all cases. The algorithm was also shown to be independent (insensitive) to the value of system parameters and the choice of initial estimates over their entire permissible range.

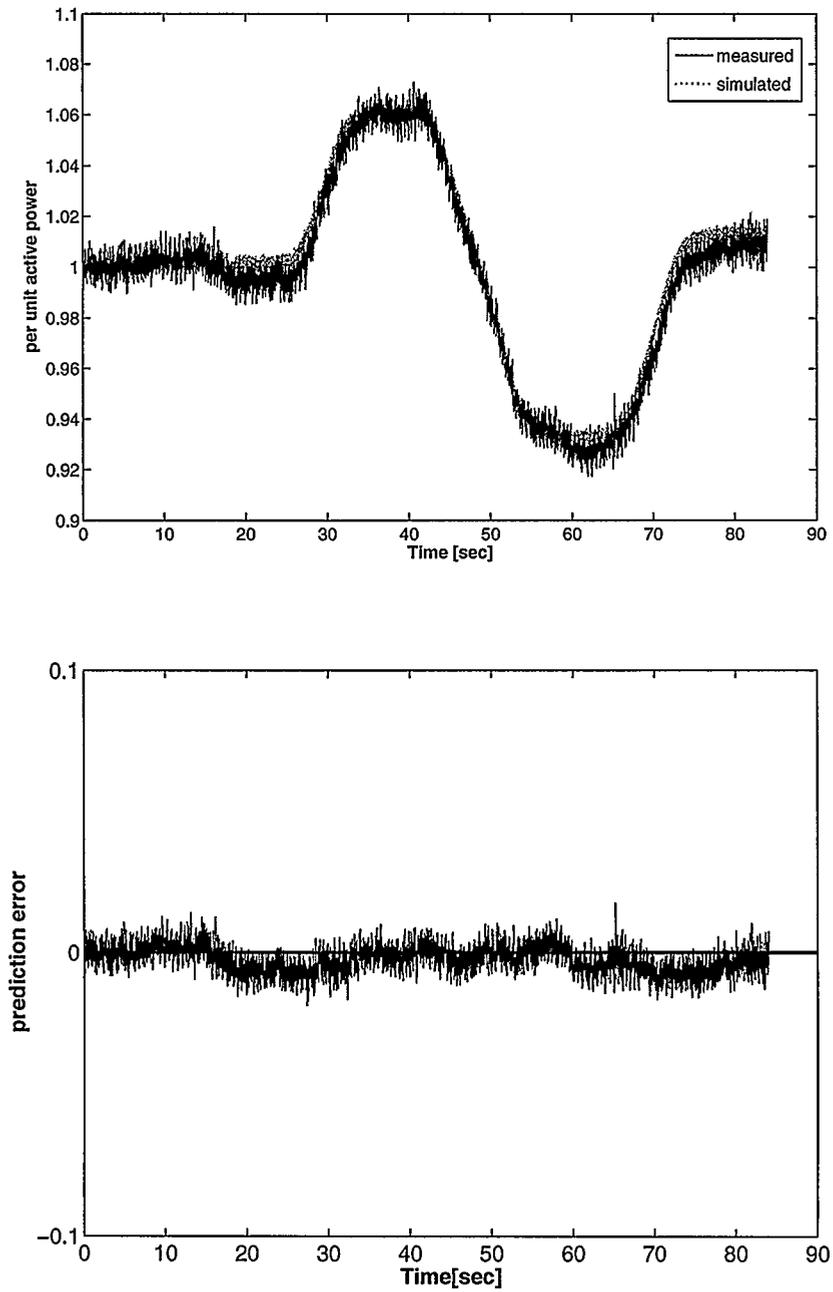


Figure 4.9: Measured and predicted active power, and prediction error, for the field data

Chapter 5

Conclusions and Summary

5.1 Overview

This thesis investigates and examines algorithms for non-convex system identification. In the context of power system load modeling and human ankle dynamics identification, the need for new algorithms is first established. More specifically, it is shown that the existing methods for power system load modeling are either very time consuming or limited in some significant aspect, while the existing methods for ankle dynamics identification require “good” initial values or do not identify the true parameters of the ankle model.

Two algorithms for non-convex identification are proposed in this thesis: a *multi-stage identification* algorithm and a *separable identification* algorithm. Then, these algorithms are applied to artificial and real power system loads in order to evaluate their performance. Also, the ankle dynamics model is identified by implementing the multi-stage algorithm.

An overview of some of the system identification and parameter estimation concepts is presented in Chapter 2. Mathematical formulation of the system identification problem, along with the non-linear model structure (NARMAX) used throughout the thesis, are discussed in detail. A few identification tools – least squares techniques and gradient descent optimization – are also demonstrated. Finally, the load model and the ankle model that are employed in this thesis are introduced.

Chapter 3 presents the proposed multi-stage algorithm for non-convex optimization problems. The algorithm is then applied to the problem of power system load modeling. More specifically, a non-linear, dynamic power system load is identified by using the proposed algorithm. The algorithm is also applied to the problem of identification of human ankle dynamics. Numerical results from artificial loads, real loads, and ankle dynamics simulations are included and discussed.

Chapter 4 provides a separable identification algorithm for non-convex optimization problems, which is subsequently applied to the non-linear, dynamic power system load model of Chapter 3. Once again, artificial and real data are used to examine the validity of the algorithm.

5.2 Contributions

The main contributions of this thesis can be summarized as:

1. A new multi-stage identification algorithm which can be applied to non-convex problems is introduced. More specifically, the power system load modeling problem and the ankle dynamics identification problem are investigated with the proposed algorithm. This technique approximates the non-linear model with a NARMAX equivalent, and then determines the optimal parameters for the NARMAX structure. Finally, these parameters are used as initial values for the non-linear optimization routine which can determine the optimal values of the original model.
2. The solution space for the dynamic power system load identification problem is explored thoroughly, and its convexity with respect to two of its parameters is

established. Two different approaches are utilized to demonstrate the convexity of the solution space:

- i) showing that the Hessian is positive definite, and
 - ii) expanding the exponential terms in the dynamic load model and reducing them to convex components.
3. Using the convexity property, a separable identification algorithm is utilized to identify the dynamic power system load. The problem is decomposed into convex and non-convex identification components. The range of the non-convex parameter is traversed using a search algorithm (Line Search, Rate of Change Search, and Improved Rate of Change Search) while a convex optimization routine is performed at each point along the path.

5.3 Conclusions

The main conclusions of this thesis are:

1. The multi-stage algorithm yields accurate results, and is capable of identifying the examined models within their permissible ranges of values.
2. For identifying power system loads, the multi-stage algorithm is much more efficient than published techniques while maintaining an equivalent level of accuracy. The proposed algorithm can also successfully determine the optimal values of a real, aggregated load.
3. The Bootstrap structure detection technique is successfully applied to the dynamic load model. The results indicate that approximating the dynamic load

model with a polynomial NARMAX model is a reasonable undertaking.

4. The power system dynamic load model is shown to be convex in two of its parameters (θ_1 and θ_2) while non-convex in the other parameter (θ_3).
5. The proposed separable algorithm achieves accurate results while converging much faster than a global search routine. As well, the algorithm is validated by simulated and field data.

Bibliography

- [1] Analysis and control of power system oscillations. CIGRE Technical Report, 1996.
- [2] R.A. Adams. *Calculus: A Complete Course*. Addison Wesley, 1999.
- [3] A.P. Alves da Silva, C. Ferreira, and G.L. Torres. Dynamic load modeling based on a nonparametric ann. *Proceedings of the ISAP International Conference on Intelligent Systems Applications to Power Systems*, pages 55–59, 1996.
- [4] A. Antoniou. *Digital Filters: Analysis and Design*. McGraw-Hill, New York, 1979.
- [5] S. Arnborg, D.J. Andersson, G. and Hill, and I.A. Hiskens. On influence of load modelling for undervoltage load shedding studies. *IEEE Transactions on Power Systems*, 13(2):395–400, May 1998.
- [6] K.J. Åström. *Computer Controlled Systems: Theory and Design*. Prentice Hall, Upper Saddle River, NJ, 1997.
- [7] E.W. Bai and D. Li. Convergence of the iterative Hammerstein system identification algorithm. *IEEE Transactions on Automatic Control*, 49(11):1929–1940, 2004.
- [8] M. Begovic and R. Mills. Load identification and voltage stability monitoring. *IEEE Transactions on Power Systems*, 10(1):109–116, February 1995.

- [9] S.A. Billings and W.S.F. Voon. Least squares parameter estimation algorithms for non-linear systems. *International Journal of System Science*, 15:601–615, 1984.
- [10] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [11] J. Bruls, C.T. Chou, B.R.J. Haverkamp, and M. Verhaegen. Linear and non-linear system identification using separable least-squares. *European Journal of Control*, 5(1):116–128, 1999.
- [12] C.A. Cañizares, I. Dobson, N. Miller, V. Ajarapu, and H. Hamadanizadeh. Voltage stability assessment: Concepts, practices, and tools. Technical Report, IEEE Power Engineering Society, Power System Stability Subcommittee, 2002.
- [13] S.J. Chapman. *Electrical Machinery and Power System Fundamentals*. McGraw-Hill Higher Education, 2001.
- [14] Byoung-Kon Choi, Hsiao-Dong Chiang, Yinhong Li, Hua Li, Yung-Tien Chen, Der-Hua Huang, and M.G. Lauby. Measurement-based dynamic load models: derivation, comparison, and validation. *IEEE Transactions on Power Systems*, 21(3):1276–1283, August 2006.
- [15] T.F. Coleman and Y. Li. On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds. *Mathematical Programming*, 67:189–224, 1994.

- [16] T.F. Coleman and Y. Li. An interior, trust-region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6:418–445, 1996.
- [17] J.E. Dennis, Jr. *State of the art in Numerical Analysis*. Academic Press, 1977.
- [18] L.G. Dias and M.E. El-Hawary. Nonlinear parameter estimation experiments for static load modelling in electric power systems. *IEE Proceedings on Generation, Transmission and Distribution*, 136(2):68–77, March 1989.
- [19] A. Gilat. *MATLAB: An Introduction with Applications*. John Wiley & Sons, 2 edition, 2004.
- [20] G.H. Golub and C.F Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, 1989.
- [21] J. Gomulka. Decomposition algorithms for smooth unconstrained optimization problems. In L.C.W. Dixon and G.P. Szego, editors, *Numerical Optimization of Dynamic Systems*, pages 269–276. North Holland, Amsterdam, 1980.
- [22] G.C. Goodwin and R.L. Payne. *Dynamic System Identification : Experiment Design and Data Analysis*. Academic, New York, 1997.
- [23] L.M. Hajagos and B. Danai. Laboratory measurements and models of modern loads and their effect on voltage stability studies. *IEEE Transactions on Power Systems*, 13(2):584–592, May 1998.
- [24] S. Haykin and B. Van Veen. *Signals and Systems*. Wiley, New York, 2002.
- [25] D.J. Hill. Nonlinear dynamic load models with recovery for voltage stability studies. *IEEE Transactions Power System.*, 8:166–176, 1993.

- [26] L. Ingber. Adaptive simulated annealing (asa): Lessons learned. *Control Cybernetica*, 25(1):33–54, 1996.
- [27] L. Jackson. *Digital Filters and Signal Processing*. Kluwer Academic Publishers, Norwell, 1995.
- [28] P. Ju, E. Handschin, and D. Karlsson. Nonlinear dynamic load modelling: model and parameter estimation. *IEEE Transactions on Power Systems*, 11(4):1689–1697, November 1996.
- [29] Wen-Shiow Kao, Chiang-Tsang Huang, and Chiew-Yann Chiou. Dynamic load modeling in taipower system stability studies. *IEEE Transactions on Power Systems*, 10(2):907–914, 1995.
- [30] D. Karlsson and D. J. Hill. Modelling and identification of nonlinear dynamic loads in power systems. *IEEE Transactions on Power Systems*, 9(1):157–166, February 1994.
- [31] R.E. Kearney and I.W. Hunter. System identification of human joint dynamics. *Critical Reviews in Biomedical Engineering*, 18(1):55–87, 1990.
- [32] R.E. Kearney, R.B. Stein, and L. Parameswaran. Identification of intrinsic and reflex contributions to human ankle stiffness dynamics. *IEEE Transactions on Biomedical Engineering*, 44(6):493–504, 1997.
- [33] V. Knyazkin, C.A. Cañizares, and L. Soder. On the parameter estimation and modeling of aggregate power system loads. *IEEE Transactions on Power Systems*, 19:1023–1031, May 2004.

- [34] V. Knyazkin, C. Cañizares, and L. Söder. On the parameter estimation of linear models of aggregate power system loads. *IEEE Power Engineering Society General Meeting*, 4:2392–2397, July 2003.
- [35] S.L. Kukreja. A suboptimal bootstrap method for structure detection of non-linear output-error models with application to human ankle dynamics. *International Journal of Control*, 78(12):937–948, 2005.
- [36] S.L. Kukreja, G.L. Galiana, and R.E. Kearney. NARMAX representation and identification of ankle dynamics. *IEEE Transactions on Biomedical Engineering*, 50:70–81, 2003.
- [37] I.J. Leontaritis and S.A. Billings. Input-output parametric models for non-linear systems part I: deterministic non-linear systems. *International Journal of Control*, 41(2):303–328, 1985.
- [38] I.J. Leontaritis and S.A. Billings. Input-output parametric models for non-linear systems part II: stochastic non-linear systems. *International Journal of Control*, 41(2):329–344, 1985.
- [39] L. Ljung. *System Identification : Theory for the user*. Prentice-Hill, Upper Saddle River, NJ, 1999.
- [40] D.W. Marquardt. An algorithm for least-squares estimation of non-linear parameters. *Journal of Society of Industrial Applied Mathematics*, 11:431–441, 1963.

- [41] M.M. Mirbagheri, H. Barbeau, and R.E. Kearney. Intrinsic and reflex contributions to human ankle stiffness: variation with activation level and position. *Experimental Brain Research*, 135(4):423–436, 2000.
- [42] L. T. M. Mota and A. A. Mota. Load modeling at electric power distribution substations using dynamic load parameters estimation. *International Journal of Electrical Power & Energy Systems*, 26:805–811, December 2004.
- [43] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, 2000.
- [44] IEEE Task Force on Load Representation for Dynamic Performance. Standard load models for power flow and dynamic performance simulation. *IEEE Transactions on Power Systems*, 10(3):1302–1313, August 1995.
- [45] A. Oonsivilai and M.E. El-Hawary. Power system dynamic load modeling using adaptive-network-based fuzzy inference system. *IEEE Canadian Conference on Electrical and Computer Engineering*, 3:1217–1222, 1999.
- [46] J. W. O’Sullivan and M. J. O’Malley. Identification and validation of dynamic global load model parameters for use in power system frequency simulations. *IEEE Transactions on Power Systems*, 11(2):851–857, May 1996.
- [47] A. Papoulis and S. Pillai. *Probability, random variables and stochastic processes*. McGraw-Hill, fourth edition, 2002.
- [48] W.W. Price. Bibliography on load models for power flow and dynamic performance simulation. *IEEE Transactions on Power Systems*, 10:523–538, February

1995.

- [49] A. Ruhe and P.A. Wedin. Algorithms for separable nonlinear least square problems. *SIAM Review*, 22:318–337, 1980.
- [50] K.R. Sales and S.A. Billings. Self-tuning control of nonlinear armax models. *International Journal of Control*, 51(4):753–769, April 1990.
- [51] R. Seydel. *Bifurcation and chaos: analysis, algorithms, applications*. Birkhäuser, 1991.
- [52] M. Taleb, M. Akbaba, and E.A. Abdullah. Aggregation of induction machines for power system dynamic studies. *IEEE Transactions on Power Systems*, 9(4):2042–2048, 1994.
- [53] C. Taylor. *Power System Voltage Stability*. McGraw-Hill, New York, 1994.
- [54] R. Vaccaro. *Digital Control: A State-Space Approach*. McGraw-Hill, New York, 1995.
- [55] M. Verhaegen. Subspace model identification part 3. analysis of the ordinary output-error state-space model identification algorithm. *International Journal of Control*, 58(3):555–586, 1993.
- [56] E. Walter and L. Pronzato. *Identification of parametric models from experimental data*. Springer, Berlin, 1997.
- [57] Jin-Cheng Wang, Hsiao-Dong Chiang, Chung-Liang Chang, Ah-Hsing Liu, Chang-Horng Huang, and Chiung-Yi Huang. Development of a frequency-

- dependent composite load model using the measurement approach. *IEEE Transactions on Power Systems*, 9(3):1546–1556, August 1994.
- [58] Jinyu Wen, Shaorong Wang, Shijie Cheng, Q.H. Wu, and D.W. Shimmin. Measurement based power system load modeling using a population diversity genetic algorithm. *Proceedings of International Conference on Power System Technology (PowerCon)*, 1:771–775, 1998.
- [59] D.T. Westwick and R.E. Kearney. Separable least squares identification of a parallel cascade model of human ankle stiffness. In *Proceedings of the IEEE EMBS Conference*, volume 23, pages 1282–1285, Istanbul, Turkey, 2001.
- [60] D.T. Westwick and R.E. Kearney. *Identification of Nonlinear Physiological Systems*. IEEE Press Series in Biomedical Engineering. IEEE Press / Wiley, Piscataway, NJ, 2003.
- [61] W. Xu, E. Vaahedi, Y. Mansour, and J. Tamby. Voltage stability load parameter determination from field tests on bc hydro’s system. *IEEE Transactions on Power Systems*, 12(3):1290–1297, August 1997.
- [62] A. Yeredor. The extended least squares criterion: minimization algorithms and applications. *IEEE Transactions on Signal Processing*, 49:74–86, 2001.
- [63] H. Yuan, D.T. Westwick, E.P. Ingenito, K. R. Lutchen, and B. Suki. Parametric and nonparametric nonlinear system identification of lung tissue strip mechanics. *Annals of Biomedical Engineering*, 27(4):548–562, 1999.
- [64] Q. Zhang, A. Iouditski, and L. Ljung. Identification of wiener system with

- monotonous nonlinearity. In *IFAC Symposium on Modelling, Identification and Signal Processing Identification and System Parameter Estimation (SYSID 2006)*, pages 166–171, Newcastle, Australia, 2006.
- [65] Yunfei Zhou, Aiguo Han, Sijie Yan, and Xuedong Chen. A fast method for online closed-loop system identification. *International Journal of Advanced Manufacturing Technology*, 31(1):78–84, November 2006.
- [66] Qi Zhu, Noriyuki Ohtsuki, Yoshikazu Miyanaga, and Norinobu Yoshida. Noise-robust speech analysis using running spectrum filtering. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 88(2):541–547, 2005.
- [67] S.Z. Zhu, Z.Y. Dong, K.P. Wong, and Z.H. Wang. Power system dynamic load identification and stability. *Proceedings of International Conference on Power System Technology (PowerCon)*, 1:13–18, December 2000.