A NEW METHOD FOR IDENTIFICATION OF MIMO HAMMERSTEIN MODEL

BY

SYED ZEESHAN RIZVI

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

JUNE 2008

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **SYED ZEESHAN RIZVI** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**.

Thesis Committee

H. Duwaish
Dr. Hussain N. Al-Duwaish (Adviser)

J. M. Bakhashwain
Dr. Jamil Bakhashwain (Member)

M. Mohandes
Dr. Mohammad Mohandes (Member)

for
Dr. Ibrahim O. Habiballah
Department Chairman

Dr. Mohammad S. Al-Homoud
Dean of Graduate Studies

21/7/08
Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dedicated to my loving Mother & Father

ACKNOWLEDGMENTS

In the name of Allah, the Most Beneficent Most Merciful

All praise is due to Allah, the source of all knowledge and strength. I acknowledge His infinite mercy and grace in making this work a success. And may His peace and blessings be upon his final messenger Muhammad, a guidance and inspiration to our lives. The successful completion of this work was made possible by a number of major contributions from different persons and organizations alike, to whom I wish to express my due gratitude.

I am grateful to King Fahd University of Petroleum & Minerals for giving me the opportunity to carry out this task under the guidance of a scholarly faculty and a proper research environment. My sincere gratitude and thanks goes to my thesis advisor Dr. Hussain N. Al-Duwaish for his meticulous attention, his guidance, and his patience with me. I am thankful to him for the multitudes of favors I have taken from him, and for all the things he has taught me. I would also like to extend my appreciation to my thesis committee members Dr. Jamil M. Bakhashwain and Dr. Mohammad Mohandes for their encouragement and cooperation.

Acknowledgment is due to my senior Muhammad Shakil for long hours of discussions on my thesis, and for his countless favors ever since I entered KFUPM. I am also thankful to Dr. Saad Azhar Ali for helping me and providing his insight on many parts of this thesis despite not being at KFUPM. Gratitude is due to Dr. Moinuddin as well, for helping me out with so many things during my early days at KFUPM.

I am grateful to my friends Saqib, Ahmed, and Salman. I will never forget the time we spent together. We certainly enjoyed doing our theses during the same time, sharing long hours of frustration and hardwork together, spending long nights keeping awake for each other just so that thesis simulations would seem less difficult. This work would not have been done so easily had it not been for their support, friendship, encouragement, and a healthy competition amongst ourselves. Special thanks to the rest of my mat'am (kitchen) members Mazhar Bhai, Babar, Asif, Ayaz, and Akhlaq Bhai for easing the load of studies with a lot of enjoyment and cooking, and for making these years one of the best years of my life.

I am thankful and obliged to my friend Ali Rasheed for his numerous sincere advices and suggestions during the last two years.

My heartfelt gratitude goes to my parents. I would never have been able to pursue this task without their cooperation and understanding. I am eternally indebted to my mother for all her prayers, concern, love and understanding. She has been my first university, and whatever knowledge or abilities I have today

have their roots in the love and teachings with which she has nurtured me. My respect and gratitude for her cannot be penned in words. I would like to especially thank my sister for her friendship and support which was both filial and maternal at the same time. Her indirect effects on me are several. Lastly, my deepest sense of gratitude to Nausheen Fatima for her endless prayers, never ending concern, and much needed support.

TABLE OF CONTENTS

LIST OF FIGURES viii

ABSTRACT (ENGLISH) xii

ABSTRACT (ARABIC) xiv

NOMENCLATURE xv

1 INTRODUCTION 1

1.1 System Identification 1

1.2 Block Oriented Approach 3

1.2.1 Hammerstein Model 4

1.2.2 Literature Review on Hammerstein Model Identification . 5

1.3 Motivation for present work 13

1.4 Thesis Contribution 14

1.5 Thesis Organization 15

2 PROPOSED IDENTIFICATION STRUCTURE 16

2.1 Problem Statement 17

2.2 Radial Basis Function Neural Network 18

2.3 State Space Model 20

2.4 Subspace Identification Method for State Space Models 21

2.4.1 Why Subspace Identification? 21

2.4.2 N4SID Numerical Algorithm 22

2.5 Particle Swarm Optimization 23

3 TRAINING ALGORITHM FOR SISO HAMMERSTEIN MODELS 26

3.1 Training Algorithm using LMS principle and Subspace Identification 26

3.2 Training Algorithm using PSO and Subspace Identification 33

4 TRAINING ALGORITHM FOR MIMO HAMMERSTEIN MODELS 36

4.1 MIMO System with Separate Nonlinearities 36

4.1.1 Training Algorithm using LMS principle and Subspace Identification 38

4.1.2 Training Algorithm using PSO and Subspace Identification 42

4.2 MIMO System with Combined Nonlinearities 43

4.2.1 Training Algorithm using LMS principle and Subspace Identification 44

4.2.2 Training Algorithm using PSO and Subspace Identification 48

5 SIMULATION RESULTS 49

5.1 Results for SISO system 49

5.1.1 Example 1: Square root nonlinearity 49

5.1.2 Example 2: Exponential/tanh nonlinearity 57

5.2 Results for MIMO system 65

5.2.1 Example 1: Exponential/tanh and Saturation Nonlinearities 65

5.2.2 Example 2: Steam Generator 75

6 CONCLUSION AND FUTURE WORK 89

6.1 Conclusions 89

6.2 Recommendation for further work 90

A PSO 91

B N4SID NUMERICAL ALGORITHM FOR SUBSPACE IDENTIFICATION **95**

B.1 Mathematical Tools and Notations 96

B.1.1 Block Hankel Matrices and State Sequences 96

B.1.2 Extended Observability Matrix 98

B.1.3 Geometric Tools 99

B.2 The Algorithm of N4SID 102

B.2.1 The Kalman State Sequences 103

B.2.2 Computing System Matrices 107

REFERENCES **109**

VITAE **121**

LIST OF FIGURES

1.1 A dynamic system with input u(t), output y(t) and disturbance e(t) 2
1.2 Block Diagram of a Hammerstein Model 4

2.1 Proposed Identification Structure 17
2.2 A three layer RBFNN with Q neurons 19
2.3 General method for All Subspace Algorithms 23
2.4 Flowchart for PSO algorithm 24

3.1 Identification of hammerstein system using LMS principle and subspace identification 30
3.2 Flowchart for SISO Hammerstein Identification Algorithm using LMS principle and Supspace Identification 32
3.3 Identification of hammerstein system using PSO and subspace identification 34
3.4 Flowchart for SISO Hammerstein Identification Algorithm using PSO and Supspace Identification 35

4.1 A P-input R-output Hammerstein Model with seperate nonlinearities 37
4.2 A P-input R-output Hammerstein Model with combined nonlinearity 43
4.3 A P-input P-output RBFNN 44

5.1 Square root nonlinearity for hammerstein model of example 1 identified by LMS based algorithm 51
5.2 Step response of linear dynamic part of hammerstein model of example 1 identified by LMS based algorithm 52

5.3 Squared error for hammerstein model of example 1 identified by LMS based algorithm 52
5.4 Step response of complete hammerstein system of example 1 identified by LMS based algorithm 53
5.5 Square root nonlinearity for hammerstein model of example 1 identified by PSO based algorithm 55
5.6 Step response of linear dynamic part of hammerstein model of example 1 identified by PSO based algorithm 55
5.7 Squared error for hammerstein model of example 1 identified by PSO based algorithm 56
5.8 Step response of complete hammerstein system of example 1 identified by PSO based algorithm 56
5.9 Exponential/tanh nonlinearity for hammerstein model of example 2 identified by LMS based algorithm 58
5.10 Step response of linear dynamic part of hammerstein model of example 2 identified by LMS based algorithm 59
5.11 Squared error for hammerstein model of example 2 identified by LMS based algorithm 59
5.12 Step response of complete hammerstein system of example 2 identified by LMS based algorithm 60
5.13 Exponential/tanh nonlinearity for hammerstein model of example 2 identified by PSO based algorithm 61
5.14 Step response of linear dynamic part of hammerstein model of example 2 identified by PSO based algorithm 62
5.15 Squared error for hammerstein model of example 2 identified by PSO based algorithm 62
5.16 Step response of complete hammerstein system of example 2 identified by PSO based algorithm 63
5.17 Saturation nonlinearity for hammerstein model of MIMO example 1 identified by LMS based algorithm 67

5.18 Exponential/tanh nonlinearity for hammerstein model of MIMO example 1 identified by LMS based algorithm 67
5.19 Step response (output y_1) of linear dynamic part of hammerstein model of MIMO example 1 identified by LMS based algorithm . . 68
5.20 Step response (output y_2) of linear dynamic part of hammerstein model of MIMO example 1 identified by LMS based algorithm . . 68
5.21 Squared output error for hammerstein model of MIMO example 1 identified by LMS based algorithm 69
5.22 Step response (outputs y_1 and y_2) of complete hammerstein system of MIMO example 1 identified by LMS based algorithm 69
5.23 Saturation nonlinearity for hammerstein model of MIMO example 1 identified by PSO based algorithm 71
5.24 Exponential/tanh nonlinearity for hammerstein model of MIMO example 1 identified by PSO based algorithm 71
5.25 Step response (output y_1) of linear dynamic part of hammerstein model of MIMO example 1 identified by PSO based algorithm . . 72
5.26 Step response (output y_2) of linear dynamic part of hammerstein model of MIMO example 1 identified by PSO based algorithm . . 72
5.27 Step response (outputs y_1 and y_2) of complete hammerstein system of MIMO example 1 identified by PSO based algorithm 73
5.28 Squared error for hammerstein model of MIMO example 1 identified by PSO based algorithm 73
5.29 Industrial Steam Generation Plant 75
5.30 Validation result for output y_1 of MIMO example 2 (steam generator) identified by LMS based algorithm 79
5.31 Validation result for output y_2 of MIMO example 2 (steam generator) identified by LMS based algorithm 79
5.32 Validation result for output y_3 of MIMO example 2 (steam generator) identified by LMS based algorithm 80

5.33 Validation result for output y_4 of MIMO example 2 (steam generator) identified by LMS based algorithm 80
5.34 Squared error for MIMO example 2 (steam generator) identified by LMS based algorithm 81
5.35 First nonlinearity for MIMO example 2 (steam generator) identified by LMS based algorithm 81
5.36 Second nonlinearity for MIMO example 2 (steam generator) identified by LMS based algorithm 82
5.37 Third nonlinearity for MIMO example 2 (steam generator) identified by LMS based algorithm 82
5.38 Fourth nonlinearity for MIMO example 2 (steam generator) identified by LMS based algorithm 83
5.39 Validation result for output y_1 of MIMO example 2 (steam generator) identified by PSO based algorithm 84
5.40 Validation result for output y_2 of MIMO example 2 (steam generator) identified by PSO based algorithm 84
5.41 Validation result for output y_3 of MIMO example 2 (steam generator) identified by PSO based algorithm 85
5.42 Validation result for output y_4 of MIMO example 2 (steam generator) identified by PSO based algorithm 85
5.43 Squared error for MIMO example 2 (steam generator) identified by PSO based algorithm 86
5.44 First nonlinearity for MIMO example 2 (steam generator) identified by PSO based algorithm 86
5.45 Second nonlinearity for MIMO example 2 (steam generator) identified by PSO based algorithm 87
5.46 Third nonlinearity for MIMO example 2 (steam generator) identified by PSO based algorithm 87
5.47 Fourth nonlinearity for MIMO example 2 (steam generator) identified by PSO based algorithm 88

THESIS ABSTRACT

NAME: Syed Zeeshan Rizvi

TITLE OF STUDY: A New Method for Identification of MIMO Hammerstein Model

MAJOR FIELD: Electrical Engineering

DATE OF DEGREE: JUNE 2008

A Hammerstein Model is composed of a static nonlinear part followed by a linear dynamic part. While identification of single input single output (SISO) hammerstein models has been dealt with efficiently, identification of multi-input multi-output (MIMO) systems is a more complex and difficult issue. In this thesis, identification is carried out by modeling the static nonlinearity with radial basis function neural network (RBFNN), while a state-space model is used to model the linear dynamic part.

Two new algorithms have been proposed in this thesis. The first algorithm makes use of least mean square (LMS) principle for identification of RBFNN weights and subspace identification for identifying state-space models. A second

algorithm uses particle swarm optimization (PSO) for estimating the weights of RBFNN and subspace identification for updating the state-space models.

For MIMO systems, update equations have been derived for two distinct cases i.e. when the nonlinearity is separate as well as for the case when the nonlinearity is combined. Simulations have been carried out and proposed algorithms have been validated.

Keywords: *Hammerstein, SISO, MIMO, RBFNN, Least Mean Square, Particle Swarm Optimization, State Space Models, Subspace Identification, Static Nonlinearity, Dynamic Linearity*

خلاصــة الرســالــة

الاسم الكامل : سيد ذيشان رضوي

عنـــــوان الرسالة : طريقة جديدة لتحديد هوية نموذج هامرستاين متعدد المدخلات والمخرجات

التخــصـــص : هندسة الكهربائية

تاريخ الشهادة : يونيو 2008

نموذج هامرستاين (Hammerstein) يتكون من جزء ساكن لاخطي يليه جزء حركي خطي. وفي حين ان التعارف على نماذج هامرستاين أحادية المدخلات والمخرجات (SISO) قد عولجت بكفاءه، فإن التعارف على النظم متعددة المدخلات والمخرجات (MIMO) هي مسألة اكثر تعقيدا وصعوبة. في هذه الرسالة، يُجرى التعارف على عبر نمذجة الجزء اللاخطي الساكن بشبكه عصبية شعاعية الأساس (RBFNN)، في حين أن نموذج الحال والفضاء (state – space) يستخدم لنمذجة الجزء الحركي الخطي.

في هذه الرسالة تم اقتراح خوارزميتين جديدتين ، الاولى تستفيد من مبدأ معدل المربعات الأدنى (LMS) لتحديد أوزان الشبكة العصبية شعاعية الأساس (RBFNN) وتستفيد من تحديد الفضاء الجزئي لتحديد نماذج الحال والفضاء (state – space). أما الخوارزميه الثانية فتستخدم سرب الجسيمات(PSO) لتقدير أوزان الشبكة العصبية شعاعية الأساس (RBFNN) وتحديد الفضاء الجزئي لتحديث نماذج الحال والفضاء (state – space).

تم اشتقاق معادلات التحديث للنظم متعددة المدخلات والمخرجات (MIMO) عند حالتين، عندما تكون اللاخطية منفصلة وعندما تكون مضمومة. وقد تمت المحاكاه والتحقق من والخوارزمية المقترحة. وقد كا نت النتا ئج مشجعة جد اً

Nomenclature

Abbreviations

SISO	:	Single Input Single Output
MIMO	:	Multi Input Multi Output
MISO	:	Multi Input Single Output
MFNN	:	Multilayer Feed Forward Neural Network
RBFNN	:	Radial Basis Function Neural Network
ARMA	:	Autoregressive Moving Average
ARMAX	:	Autoregressive Moving Average with eXternal input
GA	:	Genetic Algorithm
PSO	:	Particle Swarm Optimization
ES	:	Evolutionary Strategy
NFHM	:	Neuro Fuzzy Hammerstein Model
PAHM	:	Polynomial Approximation Hammerstein Model
LMS	:	Least Mean Square
PEM	:	Prediction Error Method
SIM	:	Subspace Identification Method
N4SID	:	Numerical Algorithm for Subspace State Space System Identification
MSE	:	Mean Squared Error

Notations

$u(t)$	:	Input to SISO hammerstein system at time t
$U(t)$	:	Input vector for MIMO hammerstein system at time t
$v(t)$	:	Output of SISO system's static nonlinearity and input to its linear dynamic part at time t
$V(t)$	:	Output vector for MIMO system's static nonlinearity and input vector to its linear dynamic part at time t
$y(t)$	:	Output of SISO hammerstein system at time t
$Y(t)$	:	Output vector for MIMO hammerstein system at time t
$\hat{y}(t)$	:	Estimated output of SISO hammerstein system at time t
$\hat{Y}(t)$	:	Estimated output vector for MIMO hammerstein system at time t
$e(t)$	:	Output error for SISO hammerstein system at time t
$E(t)$	:	Output error vector for MIMO hammerstein system at time t
I	:	Cost function
$x(t)$	:	State vector at time t
$z(t)$	:	Measurement noise at time t
$w(t)$	:	Process noise at time t
A	:	System matrix for state-space model
B	:	Input matrix for state-space model
C	:	Output matrix for state-space model
D	:	Feedthrough matrix for state-space model
η	:	Learning rate

$\phi(t)$	:	Basis function at time t
$\Phi(t)$	:	Basis function vector at time t
W_i	:	Weight vector for i^{th} nonlinearity or i^{th} nonlinear output
w_i	:	Weight corresponding to i^{th} basis function
λ_i	:	i^{th} eigen value of linear dynamic part
R^S	:	Covariance matrix
U_p	:	Block hankel matrix for past inputs
$U_{o\|i-1}$	:	Block hankel matrix for past inputs
		(Subscript indicates limits of indices for the first column of the matrix)
U_f	:	Block hankel matrix for future inputs
$U_{i\|2i-1}$	:	Block hankel matrix for future inputs
Y_p	:	Block hankel matrix for past outputs
$Y_{o\|i-1}$	:	Block hankel matrix for past outputs
Y_f	:	Block hankel matrix for future outputs
$Y_{i\|2i-1}$	:	Block hankel matrix for future outputs
Γ_i	:	Extended observability matrix
Π_A	:	Operator projecting the row space of a matrix onto the row space of A

CHAPTER 1

INTRODUCTION

1.1 System Identification

The aim of system identification is to construct mathematical models from measured input and output data. A successful identification consists of four steps namely experiment design, choice of a proper model structure, parameter estimation and model validation. Experimental design requires selection of various parameters like sampling time and excitation signal. Once these parameters are selected, the system is excited and data is obtained. The next step involves selection of a proper model structure. Actual identification experiment is then carried out and model parameters are estimated. Lastly, the identified model is verified by being excited with a fresh set of data. The generated output is compared with the measured output and in case of a mismatch between them, parameter estimation is repeated iteratively until a goal of minimum error between measured and estimated output is achieved.

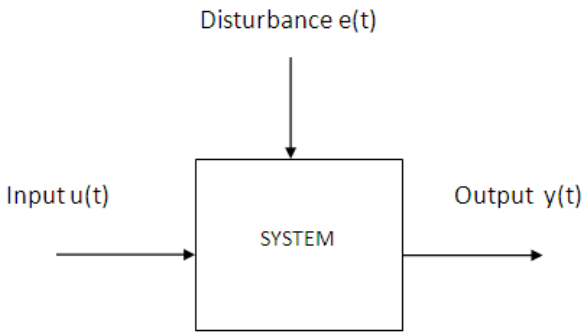

Figure 1.1: A dynamic system with input u(t), output y(t) and disturbance e(t)

System identification techniques have been extensively applied in various fields of science and engineering. For example, in the area of communications, it has been applied to effectively build reconstruction mechanisms [3]. In power systems, it has been credited to modeling of consumer loads [4]. Several studies and applications can be found in control engineering where the key goal is in output prediction for control purposes [5], [6], [7]. Another major field of science in which system identification has found rich applications is the biomedical area [8], [9].

Real life systems are usually nonlinear in nature. So in order to identify a system, usually an operating point is sought around which the system exhibits linear behavior. A lot of research has been done on system identification using linear models [10] . However, a linearized estimate does not completely model a system, and such a model may not behave perfectly outside the linear region of operation.

Nonlinear system identification has been a subject of considerable research

in the recent past [11], [12]. Over the time, several different model structures and techniques have been used to identify nonlinear systems. Genetic algorithms [40], [50] particle swarm [64], neural networks [42], adaptive filters and neuro-fuzzy networks [56], [57], [58], [82] have been used recently to identify nonlinear systems.

1.2 Block Oriented Approach

An approach that has proved very promising is based on the assumption that the identified system consists of relatively simple subsystems, and that the structure of the system is known. This is known as *Block Oriented Approach*. Based on this approach, a system can be broken down into linear and nonlinear parts, separate from each other. The subsystems are then identified on the basis of the input-output signals of the whole system and the a priori information about the system. The idea of block oriented identification is present in the literature in papers as early as those of Narendra and Gallman [27], Gardiner [14], Webb [15], Brilinger [16], Billings and Fakhouri [17].

Block oriented models have been applied to represent physical and biological systems such as a distillation column [18], [19], [20], pH control system [21], [22], an electrical generator [23], an electrical drive [24], communication bandpass circuits [25] and visual cortex [26].

1.2.1 Hammerstein Model

Block oriented identification focuses attention mainly on two types of systems, i.e., the *Hammerstein Model* and the *Wiener Model*. In the first, a nonlinear memoryless subsystem is followed by a linear dynamic one while the latter consists of the same subsystems connected in the reverse order. This thesis mainly focuses on the identification of *Hammerstein Model*.

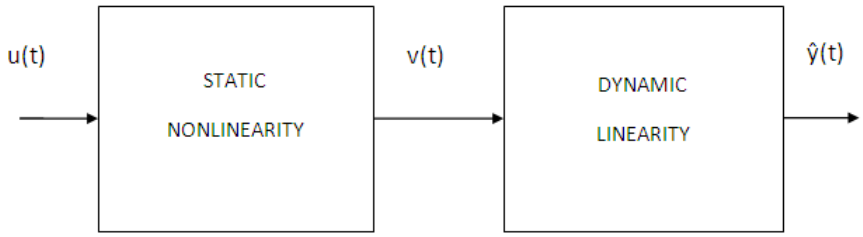

Figure 1.2: Block Diagram of a Hammerstein Model

The Hammerstein model is a very flexible representation of a non linear system as the nonlinearity is completely separate from the easily realizable linear parts. As shown in figure 1.2, the static nonlinear element scales the input *u(t)* and transforms it to *v(t)* through a non linear arbitrary function *f(u)*. The dynamics of the system are modeled by a linear transfer function, whose output is $\hat{y}(t)$.

1.2.2 Literature Review on Hammerstein Model Identification

So far a lot of attention has been paid to the identification of Hammerstein systems in the literature . In this section a detailed look is taken at the developments that have taken place in the identification of Hammerstein model over the time. While a number of identification algorithms have been proposed to identify the linear part of the Hammerstein Model, recovering the nonlinearity has appeared to be a much more difficult problem.

One of the earliest literature on Hammerstein model was apparently by Narendra & Gallman [27] in 1966, who estimated separately and sequentially the linear dynamic transfer function and the memory less nonlinear polynomial by iterative least square scheme.

Francis Chang & Luus showed in [28] in 1971 that using a non-iterative method developed by Hsia using a Hammerstein model with no zeros could be extended on to Hammerstein models having zeros with much lesser computation time than by the iterative methods proposed by Narendra & Gallman, while the accuracy of the estimates was comparable.

Billings and Fakhouri [29] in 1980 identified closed loop nonlinear systems using correlation analysis based on pseudo random inputs. They also discussed different methods of selecting pseudo random inputs.

In 1986 Greblicki and Pawlak [30], [31] presented an identification technique to identify discrete time Hammerstein model with a non parametric kernel estimate of

the regression function calculated from a dependent data. Greblicki also proposed [32] in 1989, identification of a Hammerstein functional with algorithms derived from trigonometric and Hermite orthogonal series. Greblicki and Pwlak in 1994 [33] proposed new algorithm for the identification of nonlinear subsystem in a Hammerstein model, in which the input observation is rearranged in increasing order. This results in identification of fourier coefficients of the unknown system in a certain combination of order statistics.

Sandeep, Wolodkin and Poolla [34] proposed that iterative and state space methods had up to ten times lesser error in estimating nonlinear systems than correlation methods.

Sun, Liu and Sano [35] proposed a new least squares type of identification algorithm for hammerstein models based on an over-sampling scheme. The authors claimed that by estimating the intermediate input to the linear part explicitly, one can identify an arbitrary continuous function type of nonlinear element as well as the unknown linear transfer function model. The authors proved that this method of identification showed consistent approximation of parameters of the linear part.

Zhu [37] identified a SISO Hammerstein system with a modified least squares method in 1998.

Gomez and Baeyens [38] used a numerically robust least squares estimation and single value decomposition based algorithm. They showed that using orthonormal bases to represent the linear part can help identify the linear sub system more

efficiently. Also, they claimed that using orthonormal bases functions gives the possibility of incorporating prior knowledge of the system in identification.

Westwick and Kearney [39] identified hammerstein model of an EMG using separable least squares. An iterative technique was proposed which alternatively estimated the linear element from a cross-correlation, and then fitted a polynomial to the nonlinearity via linear regression. In this way, a separable least squares optimization methods was proposed as a means of simultaneously estimating both the linear and nonlinear elements.

Hatanaka and Uosaki [40] showed that Hammerstein models can be identified using genetic programming. The unknown parameters of linear dynamic block and the nonlinear static block given by each individual were estimated with a least square method. The fitness is evaluated by AIC (Akaike information criterion). The authors claimed the usefulness of genetic programming over conventional methods for estimating the Hammerstein model.

Al-Duwaish [42], [43] proposed the use of multi-layer feed forward neural network (MFNN) and auto-regressive moving average (ARMA) model to model the static nonlinearity and dynamic linear parts of the hammerstein model respectively. The proposed method used recursive algorithm to update the weights of MFNN and parameters of ARMA.

Al-Duwaish also proposed a genetic approach to identify hammerstein models in [44]. Parameters of the hammerstein models were estimated using genetic algorithms from the input-output data by minimizing the error between the outputs

of the true and identified models. The author was also able to identify systems having non-minimum phase characteristics.

Marchi and Coelho [45] presented a comprehensive comparison of several parametric and structural methods of nonlinear system identification. Among the methodologies compared were linear mathematical models, volterra models, hammerstein models, and bilinear models. Two different structural models consisting of neural networks were used, one employing multilayer perceptron and the other using radial basis function. Results were validated and compared on a fan and plate process. The results showed that bilinear and hammerstein models provided a better tracking of the output than linear model. Structural models were found to be more computationally complex but showed better performance than parametric models. Radial basis function approximated the nonlinearities in the system better than the multilayer perceptron.

Darouach and Boutayeb [46] proposed a simple method for recursive identification of multi-input single-output (MISO) Hammerstein model in the presence of unknown but bounded disturbances. The authors claimed that the estimated parameters were consistent with the measurements and the noise constraints. The authors used Lyapunov approach to enhance convergence.

Al-Duwaish and Saad Azhar Ali [47] identified Wiener and Hammerstien models using Radial Basis Neural Network, and ARMA model.

In 2001, Hassouna and Ouvrard [48] presented an identification scheme using volterra series expansion. Identification model consisted of a volterra series with

its first term truncated. Volterra kernels were expanded on multidimensional orthonormal bases. The authors claimed accurate approximation of nonlinear systems, provided the nonlinearities were continuous and smooth.

Bai used a blind approach for hammerstein model identification in [49]. It was shown that identification of linear part can be can be achieved based only on the output measurements. This meant that hammerstein model identification could be made possible without knowing the structure of the nonlinearity and the internal variable. Nonlinear part was estimated easily once linear part was approximated accurately.

A. Akramizadeh, A. Ali and K. Hamid proposed using Genetic Algorithm for the identification of a Hammerstein Model in [50]. The author proposed Genetic Algorithm to find out the nonlinear function parameters as well as the poles and zeros of the linear function. The authors also proposed a new method to increase the speed of the search by introducing a *dynamic mutation rate*. The authors parameterized the linear part as an ARMA model and used evolutionary LMS algorithm.

Luo and Leonessa [51] identified MIMO nonlinear dynamic systems containing the cascade of a linear dynamic system with static nonlinearities at both the input and feedback loop. A subspace instrumental variable identification method was used for the closed loop system. The authors claimed accurate identification results using numerical examples.

Kozek and Jovanovic [52] identified Hammerstein Model using Extended

Kalman Filters, which was used for parameter identification of Hammerstein systems. For efficient estimation of unknown nonlinearities, linear parametrizations with linear static mappings and basis function expansions were proposed and the EKFs for these cases were established. The authors claimed efficiency and validated their proposal through simulation results.

Hatanaka and Uosaki [53] proposed identification by Evolutionary Computational Approach like Genetic Algorithm (GA) or Evolutionary Strategies (ES).

Voros [41] showed that hammerstein models with discontinuous nonlinearities containing dead zones can be estimated using recursive least squares identification supplemented with estimation of model internal variables. The author claimed that this method could be extended to other types of static nonlinearities like multisegment piecewise linear nonlinearities.

Bai [54] showed that the linear part can be decoupled from the nonlinear part in hammerstein model identification. Therefore, identification of the linear part for a Hammerstein model becomes a linear problem and accordingly enjoys the same convergence and consistency results as if the unknown nonlinearity is absent.

A. Janczak [55] proposed four different gradient calculation algorithms for identification of neural network based hammerstein models. Besides these, other algorithms that combined steepest descent algorithm with recursive least squares (RLS) were also proposed. Computation complexity of these algorithms were analyzed and compared.

In 2004, Chiu, Jia and Ge [56] proposed a Neuro-Fuzzy based method for

hammerstein identification. The authors proposed a similar algorithm [57] for hammerstein model identification, but used an Adaptive method for that. The algorithm was developed using Lyapunov Stability theory and the authors claimed that the algorithm showed better results than a simple Neuro-Fuzzy method.

Neuro-fuzzy hammerstein model (NFHM) and Polynomial approximation hammerstein model (PAHM) approaches were used to identify a hammerstein model in [58]. The first approach used a fuzzy model while the second approach used a polynomial to estimate the parameters of static nonlinearity. Both approaches used least squares parameter estimation to estimate parameters of the linear dynamic part. Promising results were shown by identifying a complex nonlinear system.

Hhachino, Deguchi and Takata [59] proposed an identification method using Radial Basis Neural Network and Genetic Algorithm (GA). Unknown nonlinear static part was estimated by an RBF network. The weighting parameters of the RBF network and the system parameters of the linear dynamic part were estimated by the linear least-squares method. The adjusting parameters for the RBF network structure, i.e. the weights, centers and widths of the RBF were properly determined using Genetic Algorithm.

In 2005, Goethals, Pelckmans, Suykens and De Moor [60] used Subspace Identification and Least Squares Support Vector Machines to identify Hammerstein Systems. The authors reduced this to a constrained optimization problem, which was solved with Lagrangian Multiplier. The algorithm was then extended to

Hammerstein FIR models.

Chen and Hu [61] identified hammerstein systems using their own nonparametric approach. The approach used stochastic approximation.

Greblicki in 2006 [62] again proposed a method for Continuous-Time Hammerstein Identification from sampled data. A continuous-time Hammerstein system was driven by a random signal and identified from observations sampled in time.

Westwick [63] presented another approach for identification of hammerstein systems. This time it dealt with time varying hammerstein models, i.e. hammerstein systems in which parameters of linear and nonlinear parts varied with time. The algorithm expanded the system's time varying parameters into finite sets of basis sequences transforming the identification problem into a time-invariant one. Results were obtained from a simulation study of a time varying hammerstein system, and algorithm was verified.

Zang & Liu [64] used PSO for identification of hammerstein models.

In 2007, Wenxiao [65] identified hammerstein system using extended least squares Algorithm. The nonlinear static function was expressed as a linear combination of basic functions with unknown coefficients. The author claimed strong consistency of the estimates and good convergent rates.

Hong & Mitchel [66] identified a Hammerstein System using Bezier-Bernstein approximation. The nonlinear static function was characterized using this approximation. The identification method was based on a hybrid scheme including least squares algorithm and $Gauss - Newton$ algorithm subject to constraints.

1.3 Motivation for present work

For the identification of a multi-input multi-output (MIMO) system, computational complexity is an overwhelming issue. As a MIMO system has complex internal structure, parameterization becomes complicated and affects the estimation of all polynomial models. To estimate a polynomial model for a MIMO system, a particular model parameterization is needed first. Furthermore, to make the iterative optimization successful, one might have to provide an accurate initial estimate. However, providing any of these two prerequisites might be hard.

On the other hand, a state-space model represents an unknown system with state vectors. One only needs to provide the order of a system, which can come from prior knowledge of the system or which can be obtained by calculating singular values of an information matrix. After choosing the system order, a one-time algorithm can be implemented to estimate all the parameters without involving complex iterative nonlinear optimization. Therefore state-space models are by far more suitable for identifying a MIMO system.

The advantages of a state-space model include: *only one structural decision to make, no iterative optimization involved,* and *less computational complexity.*

The motivation for present work stems from the above stated ease and efficiency of modeling a MIMO linear system by a state-space model. This thesis proposes a method for identification of MIMO hammerstein systems by modeling its linear dynamic part with a state-space model.

1.4 Thesis Contribution

This thesis proposes a novel method for identification of MIMO Hammerstein Systems. The linear dynamics of the hammerstein system are modeled using state-space models and the non linearities are modeled by *radial basis function neural network* (RBFNN). An algorithm based on *least mean square* (LMS) principle and *subspace identification* is derived. Another algorithm is derived which is based on *particle swarm optimization* (PSO) and *subspace identification*.

Total contributions of this thesis includes:

- Development of training algorithm for a general hammerstein system using *least mean squares* (LMS) principle and *subspace identification*. Then based on this algorithm,
  - Derivation of update equation for SISO hammerstein system.
  - Derivation of update equation for MIMO hammerstein system with separate non linearities.
  - Derivation of update equation for MIMO hammerstein system with combined non linearities.
- By using *particle swarm optimization* (PSO) and *subspace identification*,
  - Development of training algorithm for SISO hammerstein system.
  - Development of training algorithm for MIMO hammerstein system with separate non linearities.

  - Development of training algorithm for MIMO hammerstein system with combined non linearities.

- Validation of above mentioned algorithms by means of implementation on different systems and examples.

1.5 Thesis Organization

This thesis has been organized in the following manner. Chapter 2 describes the proposed identification structure, and gives details of all the models and tools used in the thesis, namely *State-space Models, Subspace Identification for State-space Models, Radial Basis Function Neural Network* (RBFNN), and *Particle Swarm Optimization* (PSO).

Chapter 3 describes the proposed algorithm for identification of SISO systems, and takes a look at the mathematical working to obtain update equations for LMS based algorithm. PSO based algorithm is also investigated.

Chapter 4 describes the proposed algorithm for identification of MIMO systems, and takes a look at the mathematical working to obtain update equations for LMS based algorithm. PSO based algorithm is also investigated and compared.

In chapter 5, these algorithms are validated by carrying out simulations on different examples and systems. These simulation results are analyzed, and the results are investigated.

Finally, in chapter 6, a conclusion is drawn and possibilities of future enhancements are discussed.

CHAPTER 2

PROPOSED IDENTIFICATION STRUCTURE

The proposed identification structure consists of a *Radial Basis Function Neural Network* (RBFNN) for modeling the static nonlinearity of the system, and a *State Space Model* for modeling the linear dynamic part.

In the identification of hammerstein model, it is desired that the error between the actual output and the estimated output be minimized for accurate identification. Therefore, in a way this becomes an optimization problem where a cost index is to be minimized. *Least mean square* (LMS) principle is therefore used to train the RBFNN for minimum error. *Particle Swarm Optimization* (PSO) is also used for the same purpose. *Subspace identification* is used to estimate the system matrices of the state space model.

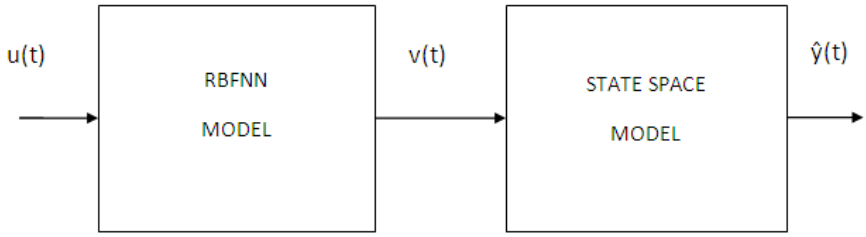

Figure 2.1: Proposed Identification Structure

2.1 Problem Statement

The hammerstein model identification problem therefore becomes the following: Given a set of (N) noisy inputs $u(t)_{t=0}^{N-1}$ and outputs $y(t)_{t=0}^{N-1}$

- Find the weights of RBF neural network and
- Find the matrices of the state space system

Since the output *y(t)* is nonlinear in relation to the input *u(t)*, the problem is nontrivial. Therefore, a recursive algorithm is to be developed to update the weights of the neural network for each set of input and output data. LMS principle is therefore used, which is one of the famous gradient methods found in the literature where a pre defined cost index is minimized, and a minimum is sought in the negative direction of the gradient of that cost index. PSO is also used to train the neural network because of its efficient optimization capabilities.

2.2 Radial Basis Function Neural Network

Radial Basis Function Neural Network is a type of neural network that uses radial basis function as activation functions. They are ideal for applications like function approximation and pattern recognition. In these networks, learning usually involves only one layer with lesser computations. Therefore, the training time for RBFNN is much less than that of MFNN [67].

These features make RBFNN attractive in many practical problems. RBFNN model in its most basic form consists of three layers. An input layer that connects the network to its environment. The second layer, often called the hidden layer performs a fixed nonlinear transformation with no adjustable parameters. The output layer then implements a linear weightage and supplies the response of the network to the output. Thus, the only adjustable parameters are the weights of this layer [67], [68]. Figure 2.2 shows a basic RBFNN with Q neurons in the hidden layer.

Each of the input node is connected to all the nodes in the hidden layer through unity weights. The nodes of the hidden layer however are connected to the output node through some weights. Every neuron performs a nonlinear transformation of the distance between the input and the center of the neuron. The nonlinearity within an RBF network can be chosen from a few typical nonlinear functions. The radial basis function has a maximum of 1 when its input is 0. As the distance between the input and the center of the neuron decreases, the output increases. Thus, a radial basis neuron acts as a detector that produces 1 whenever the input

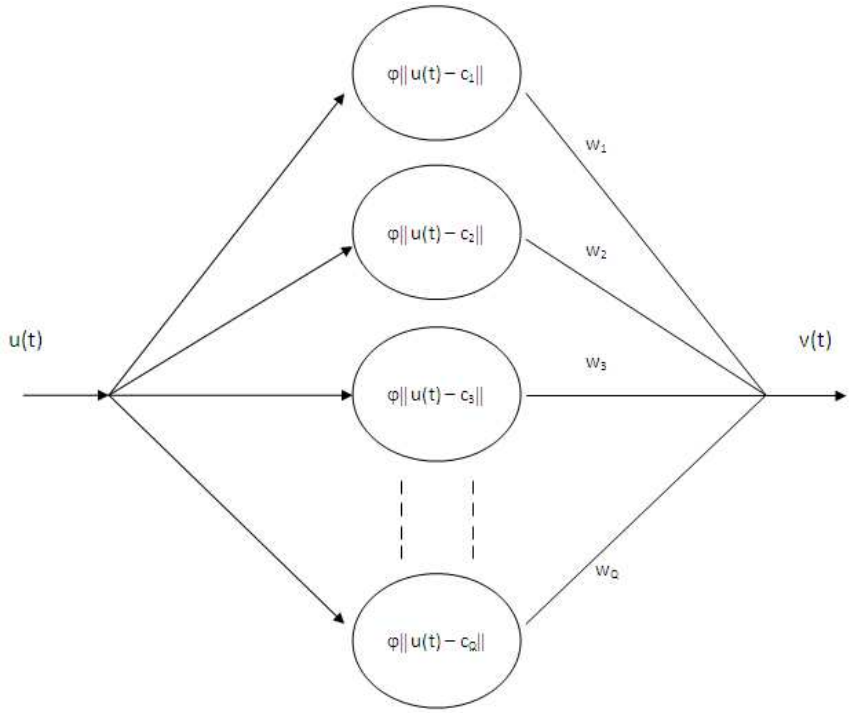

Figure 2.2: A three layer RBFNN with Q neurons

is identical to the center of the neuron.The output of an RBFNN has the following general form

$$v(t) = \sum_{i=1}^{Q} w_i \phi \|u(t) - c_i\| \tag{2.1}$$

where Q is the number of neurons in the hidden layer, c_i is the chosen center for each neuron, w_i is the corresponding weight that connects the particular neuron node to the output layer, ϕ is an arbitrary generally nonlinear function known as radial basis function, and $\|.\|$ denotes the norm that is usually Euclidean [67]. Therefore we can write

$$V(t) = W\Phi(t)$$

where

$$W = [w_1 \quad w_2 \quad w_3 \quad \cdots \quad w_Q]$$

$$\Phi(t) = [\phi\|u(t) - c_1\| \quad \phi\|u(t) - c_2\| \quad \cdots \quad \phi\|u(t) - c_Q\|]^T$$

2.3 State Space Model

As mentioned above, the proposed identification structure models the dynamic linear part with the famous *state-space model*. The equations for a discrete time state-space representation are given by

$$x(t+1) = Ax(t) + Bu(t) + w(t) \tag{2.2}$$

$$y(t) = Cx(t) + Du(t) + z(t) \tag{2.3}$$

where $u(t) \epsilon \Re^m$ and $y(t) \epsilon \Re^l$ are the vectors for m inputs and l outputs of the system observed at discrete time instant t. Vector $z(t) \epsilon \Re^{l\times 1}$ is called the measurement noise and $w(t) \epsilon \Re^{n\times 1}$ is called the process noise.

Matrix $A \epsilon \Re^{n\times n}$ is called the system matrix of order n. It describes the dynamics of the system. $B \epsilon \Re^{n\times m}$ is the input matrix which represents the linear transformation by which the deterministic inputs influence the next state. $C \epsilon \Re^{l\times n}$ is the output matrix which describes how the internal state is transferred

to the outside world. The term with the matrix $D \epsilon \Re^{l \times m}$ is called the direct feedthrough term.

2.4 Subspace Identification Method for State Space Models

2.4.1 Why Subspace Identification?

One of the earliest publications on system identification was by *Aström* and Bohlin in 1965 [1], which proposed a stochastic method known as the *maximum likelihood method* to estimate ARMAX models. Many papers have been published after that on various stochastic methods, most of which fall under the category of *Prediction Error Methods* (PEM). This has resulted in a well established theory for identification of SISO systems.

Identification of MIMO systems however is a problem which has not been dealt with satisfactorily by PEM where identification is carried out on the basis of minimization of predicted error. Identification based on PEM is a complicated function of the system parameters, and has to be solved by iterative descent methods which may get stuck into local minima. Moreover, optimization methods need an initial estimate for a canonical parametrization model, which might not be easy to provide. PEM have therefore inherent difficulties for MIMO systems.

Around 1965, Ho and Kalman published their paper [2] and introduced the state-space representation. The solution of this deterministic realization problem

was later extended by Akaike and others in 1974 to stochastic realizations. This technology was extended in the nineties to subspace state-space identification.

Subspace identification methods (SIM) do not need nonlinear optimization techniques, nor does it need to impose to the system a canonical form. Subspace methods therefore do not suffer from the inconveniences encountered in applying PEM methods to MIMO system identification.

2.4.2 N4SID Numerical Algorithm

In the category of subspace identification schemes, the most commonly used algorithm is *Numerical algorithm for Subspace State Space System Identification* (N4SID) . It was proposed by Peter Van Overschee and De Moor [70], [71] in 1994.

For a large number of input $u(t) \epsilon \Re^m$ and output $y(t) \epsilon \Re^l$ measurements generated by the unknown system of eqs. 2.2 and 2.3, the objective is to determine the order n of the system, the system matrices $A \epsilon \Re^{n \times n}$, $B \epsilon \Re^{n \times m}$, $C \epsilon \Re^{l \times n}$ and $D \epsilon \Re^{l \times m}$ and if required, the Kalman gain matrix K, without any prior knowledge of the structure of the system. Figure 2.3 shows a block diagram representation of this algorithm. On the left hand side, subspace method approach is shown where the kalman filter states are first estimated directly from input and output data, then the system matrices are obtained. The right hand side shows the classical approach in which the system matrices are found first and then the estimate of the states. Mathematical details of N4SID algorithm are described in Appendix

A of this thesis.

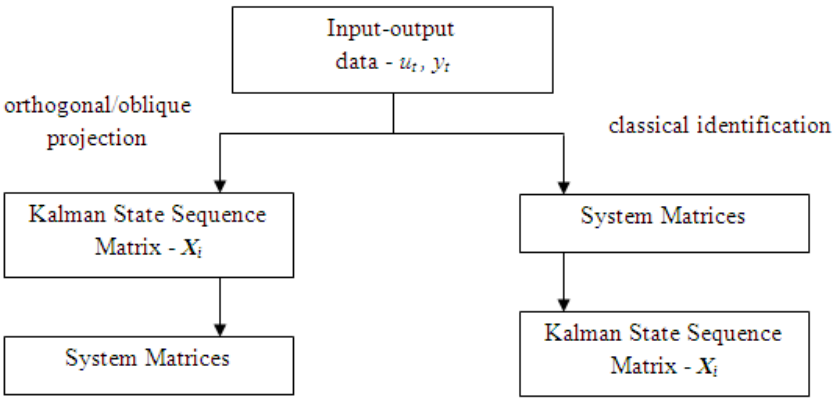

Figure 2.3: General method for All Subspace Algorithms

2.5 Particle Swarm Optimization

Particle Swarm Optimization is one of many heuristic search based optimization technique inspired by biological behavior of organisms in nature. It was first proposed and developed by Kennedy and Eberhart [79]. It is a technique based on *swarm intelligence* method [78].

PSO is inspired by behavior of a swarm of several particles like *school of fish* or *flock of birds*, where each particle position indicates the behavior of an individual fish or bird. Each particle position can be thought of as being a state of mind of that particular individual. This state of mind represents the individual's beliefs and attitudes. The movement of the particles thus correspond to the change of these beliefs. Swarms or social groups adjust their beliefs and attitudes by eval-

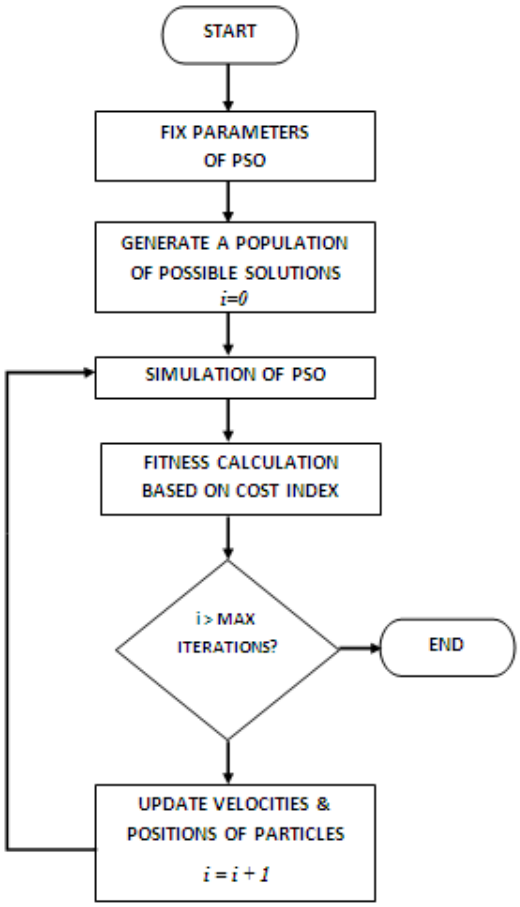

Figure 2.4: Flowchart for PSO algorithm

uating the stimuli from the environment around them and comparing it to their existing knowledge. If these stimuli or values are found to be more fit, they replace their existing values. These three properties of animal social behavior i.e., evaluation, comparison, and imitation, are what forms the basis for the particle swarm optimization algorithm. PSO uses these concepts in solving complex minimization problems.

PSO is closely related to swarm intelligence where there is no central control and no one gives orders, rather each particle acts as an individual upon local information. The swarm, as a whole, is able to perform tasks much more complex than an individual can do. The interactions among the particles makes it possible for the whole swarm to perform optimization of complex functions.

Figure 2.4 shows the basic algorithm of PSO. In this thesis, PSO is used to search for optimum weights of the RBFNN output layer. Details of the algorithm can be found in appendix A of this thesis.

CHAPTER 3

TRAINING ALGORITHM FOR SISO HAMMERSTEIN MODELS

3.1 Training Algorithm using LMS principle and Subspace Identification

This section discusses the training algorithm developed for identification of a single input single output (SISO) system. The algorithm is based on the famous *least mean squares* principle. LMS principle was first proposed by Widrow and Hoff in [81]. It is a type of gradient search technique known as the steepest descent algorithm in which the weights of the nonlinear function are updated in the negative direction of the gradient of a cost function. Ideally, the identified system is supposed to give an output identical to the output of the actual system,

but in reality, this does not happen. Therefore an optimization criterion such as the squared error is sought to be minimized.

$$\begin{aligned} I &= e^2(t) \\ &= [y(t) - \hat{y}(t)]^2 \end{aligned} \tag{3.1}$$

where $y(t)$ is the actual output and $\hat{y}(t)$ is the estimated output at time t. Hence, the weights of the nonlinearity are updated at every data point in the negative direction of the gradient of equation 3.1 [81], [67].

As mentioned in the identification structure proposed in the previous chapter, the static nonlinearity is modeled by an RBFNN which is followed by a dynamic linear part modeled by a state-space model. Hence the objective is to develop an algorithm, which updates not only the weights of the RBFNN, but also the matrices of the state space model. Weights of the RBFNN are updated using the LMS principle while supspace identification is used for identifying the matrices of the state space model.

The parameters of the nonlinearity (weights of the RBFNN) are updated by minimizing the cost index given in equation 3.1. The weights are adjusted in the negative direction of the gradient of the cost function as

$$W(k+1) = W(k) - \eta \frac{\partial I}{\partial W} \tag{3.2}$$

where η is the learning rate of the RBFNN, and is usually a small number, W ϵ

$\Re^{1\times Q}$ is the weight vector for RBFNN and is given by

$$W = [w_1 \quad w_2 \quad w_3 \quad \cdots \quad w_Q]$$

where Q is the number of neurons in the network. Using equation 3.2, we derive the partial derivatives of the cost function *w.r.t.* W as

$$\begin{aligned}
\frac{\partial I}{\partial W} &= \frac{\partial e^2(t)}{\partial W} \\
&= 2e(t)\frac{\partial e}{\partial W} \\
&= 2e(t)\frac{\partial}{\partial W}\left[y(t) - \hat{y(t)}\right] \\
&= 2e(t)\left[\frac{\partial y(t)}{\partial W} - \frac{\partial \hat{y(t)}}{\partial W}\right] \\
&= -2e(t)\frac{\partial \hat{y}(t)}{\partial W} \\
&= -2e(t)\frac{\partial}{\partial W}[Cx(t) + Dv(t)] \\
&= -2E(t)^T\frac{\partial}{\partial W_i}\left[C\{Ax(t-1) + Bv(t-1)\} + Dv(t)\right]
\end{aligned}$$

$$
\begin{aligned}
&= -2e(t)\frac{\partial}{\partial W}\left[C\left(Ax(t-1)+Bv(t-1)\right)+Dv(t)\right] \\
&= -2e(t)\left[\frac{\partial CAx(t-1)}{\partial W}+\frac{\partial CBv(t-1)}{\partial W}+\frac{\partial Dv(t)}{\partial W}\right] \\
&= -2e(t)\left[\frac{\partial CBv(t-1)}{\partial W}+\frac{\partial Dv(t)}{\partial W}\right] \\
&= -2e(t)\left[\frac{\partial CBf\left(u,t-1\right)}{\partial W}+\frac{\partial Df\left(u,t\right)}{\partial W}\right] \\
&= -2e(t)\left[\frac{\partial CBW\Phi(t-1)}{\partial W}+\frac{\partial DW\Phi(t)}{\partial W}\right]
\end{aligned}
$$

where

$$W = [w_1 \quad w_2 \quad w_3 \quad \cdots \quad w_Q]$$

$$\Phi(t-1) = [\phi\|u(t-1)-c_1\| \quad \phi\|u(t-1)-c_2\| \quad \cdots \quad \phi\|u(t-1)-c_Q\|]^T$$

$$\Phi(t) = [\phi\|u(t)-c_1\| \quad \phi\|u(t)-c_2\| \quad \cdots \quad \phi\|u(t)-c_Q\|]^T$$

$$\Rightarrow \frac{\partial I}{\partial W} = -2e(t)\{CB\Phi(t-1)+D\Phi(t)\}$$

$$\boxed{W(k+1) = W(k)+2e(t)\eta\{CB\Phi(t-1)+D\Phi(t)\}} \tag{3.3}$$

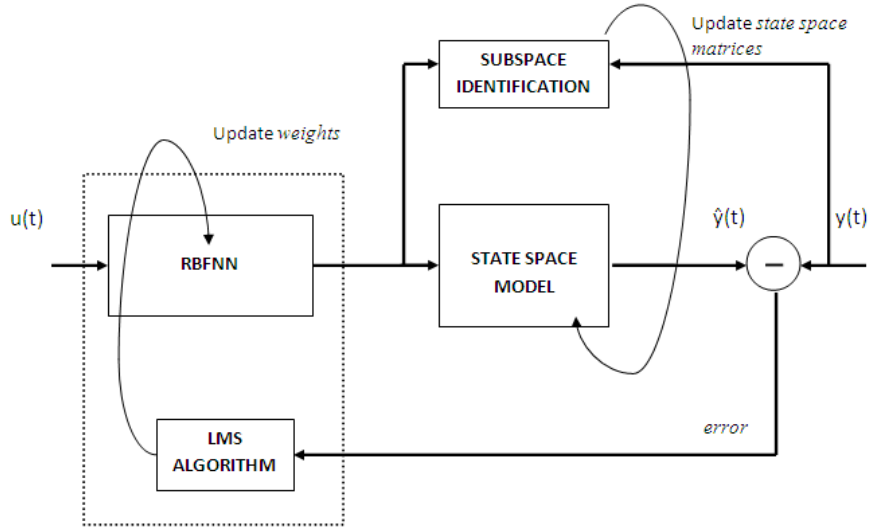

Figure 3.1: Identification of hammerstein system using LMS principle and subspace identification

The developed algorithm can be summarized as follows

1. Initialization of neural network with random weights.

2. Estimation of state space matrices A,B,C *and* D from original non linear data using subspace identification. This would be an initial estimate.

3. Generation of output $\hat{y(t)}$ from the estimated system.

4. Calculation of error and updation of weights according to equation (3.3)

5. Estimation of neural network outputs $v(t)$ once the weights of the neural network are updated for all data points.

6. Estimation of state space matrices *A,B,C and D* from the new output of neural network $v(t)$ (which is also the input to the linear system) and original system output $y(t)$. This estimate of linear system would be much closer to the original system than the previous estimate.

7. Regeneration of output $\hat{y(t)}$ from the new estimate of the system.

8. If the *mean square error* between *y(t)* and $\hat{y(t)}$ is not less than the required goal, repetition of steps 4 to 8.

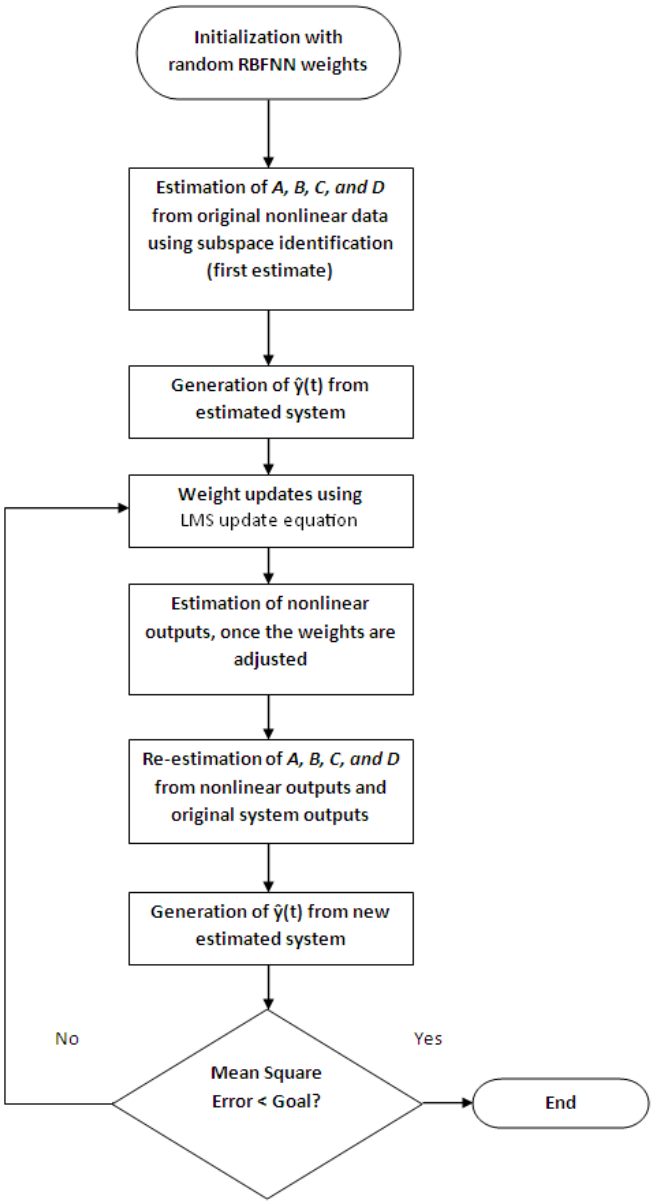

Figure 3.2: Flowchart for SISO Hammerstein Identification Algorithm using LMS principle and Supspace Identification

3.2 Training Algorithm using PSO and Subspace Identification

As noted above, for the identification of hammerstein model, we are keen to minimize error between the desired output and the observed output as given in equation 3.1.

In this second approach, particle swarm optimization is used instead of the least mean square principle for optimization of our cost function, and to find optimum weights of nonlinearity. PSO performs a heuristic search on a possible population of nonlinearity weights. The smaller the output error, the greater is the fitness of the particle. Based on this principle, the algorithm can be summarized in the following steps

1. Estimation of state space matrices *A,B,C and D* from original non linear data using subspace identification. This would be an initial estimate.

2. Initialization of PSO to find optimum weights for the neural network.

3. Obtaining a global best set of weights which minimizes the cost index given in equation 3.1.

4. Estimation of neural network outputs $v(t)$ once optimum weights of neural network are obtained.

5. Estimation of state space matrices *A,B,C and D* from the new output of neural network $v(t)$ (which is also the input to the linear system) and original

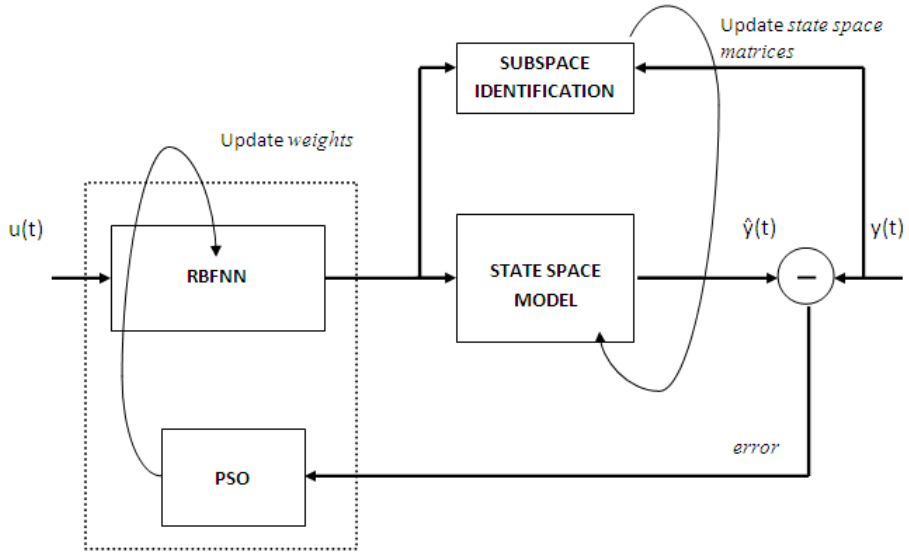

Figure 3.3: Identification of hammerstein system using PSO and subspace identification

system output $y(t)$. This estimate of linear system would be much closer to the original system than the previous estimate.

6. Regeneration of output $\hat{y(t)}$ from the new estimate of the system.

7. If the *mean square error* between $y(t)$ and $\hat{y(t)}$ is not less than the required goal, repetition of steps 2 to 6.

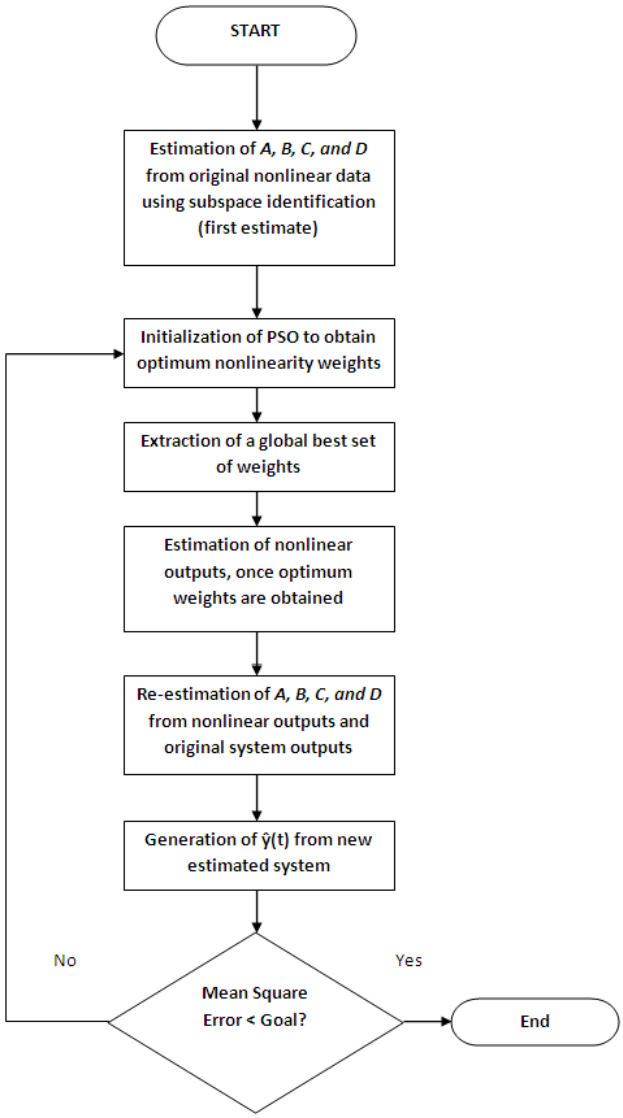

Figure 3.4: Flowchart for SISO Hammerstein Identification Algorithm using PSO and Supspace Identification

CHAPTER 4

TRAINING ALGORITHM FOR MIMO HAMMERSTEIN MODELS

This chapter discusses the training algorithm developed for identification of a multi input multi output (MIMO) system. In MIMO hammerstein systems, static nonlinearity can either be separate or combined. The following subsections take a look at both of them one by one.

4.1 MIMO System with Separate Nonlinearities

Consider a P input, R output general hammerstein system. The system has P nonlinearities separate from each other. These nonlinearities have P outputs. The linear part has R outputs. Therefore inputs $U(t)\ \epsilon\ \Re^{P\times 1}$, nonlinear outputs $V(t)$

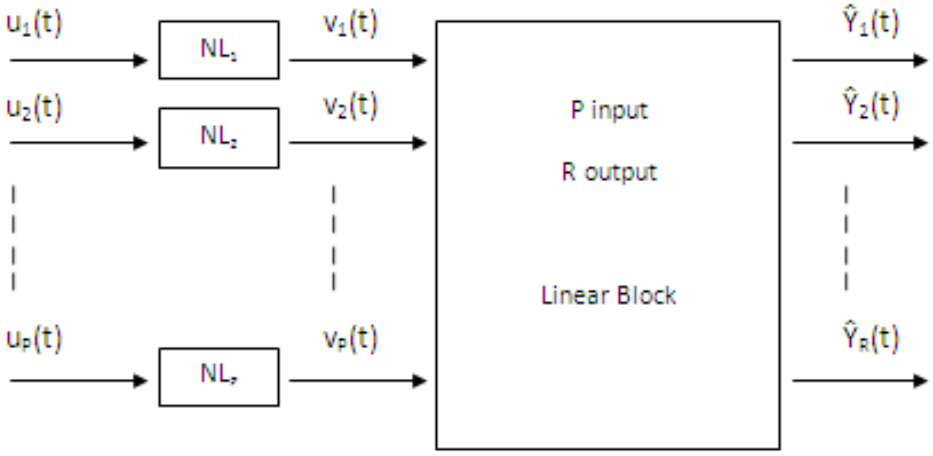

Figure 4.1: A P-input R-output Hammerstein Model with seperate nonlinearities

$\epsilon\ \Re^{P\times 1}$, and system outputs $\hat{Y}(t)\ \epsilon\ \Re^{R\times 1}$ are given by

$$U(t) = [u_1(t) \quad u_2(t) \quad \cdots \quad u_P(t)]^T \tag{4.1}$$

$$V(t) = [v_1(t) \quad v_2(t) \quad \cdots \quad v_P(t)]^T \tag{4.2}$$

$$\hat{Y}(t) = [\hat{y}_1(t) \quad \hat{y}_2(t) \quad \cdots \quad \hat{y}_R(t)]^T \tag{4.3}$$

Error between desired and observed outputs $E(t)\ \epsilon\ \Re^{R\times 1}$ is given by

$$\begin{aligned} E(t) &= [(y_1(t) - \hat{y}_1(t)) \quad (y_2(t) - \hat{y}_2(t)) \quad \cdots \quad (y_R(t) - \hat{y}_R(t))]^T \\ &= [e_1(t) \quad e_2(t) \quad \cdots \quad e_R(t)]^T \end{aligned} \tag{4.4}$$

4.1.1 Training Algorithm using LMS principle and Subspace Identification

Like its SISO counter part derived earlier, this algorithm is also based on *least mean square* principle. The cost function I is given by

$$I = E(t)^T E(t) \tag{4.5}$$

For P inputs and R outputs, the dimensions of the state space matrices for an n^{th} order system will be $A \epsilon \Re^{n \times n}$, $B \epsilon \Re^{n \times P}$, $C \epsilon \Re^{R \times n}$, and $D \epsilon \Re^{R \times P}$

For P inputs, There are P separate nonlinearities. The weights W_i of the RBFNN for i^{th} particular nonlinearity are updated in the negative direction of the gradient of cost function given in equation 4.5

$$W_i(k+1) = W_i(k) - \eta \frac{\partial I}{\partial W_i} \tag{4.6}$$

Now finding the partial derivative of eq 4.5 $w.r.t$ W_i, corresponding to the i^{th} nonlinearity

$$\begin{aligned} \frac{\partial I}{\partial W_i} &= \frac{\partial}{\partial W_i} E(t)^T E(t) \\ &= 2E(t)^T \frac{\partial}{\partial W_i} E(t) \end{aligned}$$

$$
\begin{aligned}
&= 2E(t)^T \frac{\partial}{\partial W_i} E(t) \\
&= 2E(t)^T \frac{\partial}{\partial W_i} \left[(y_1(t) - \hat{y}_1(t)) \quad (y_2(t) - \hat{y}_2(t)) \quad \cdots \quad (y_R(t) - \hat{y}_R(t)) \right]^T \\
&= 2E(t)^T \frac{\partial}{\partial W_i} \left[-\hat{y}_1(t) \quad -\hat{y}_2(t) \quad \cdots \quad -\hat{y}_R(t) \right]^T \\
&= -2E(t)^T \frac{\partial}{\partial W_i} \left[Cx(t) + DV(t) \right] \\
&= -2E(t)^T \frac{\partial}{\partial W_i} \left[C\{Ax(t-1) + BV(t-1)\} + DV(t) \right] \\
&= -2E(t)^T \left[\frac{\partial CAx(t-1)}{\partial W_i} + \frac{\partial CBV(t-1)}{\partial W_i} + \frac{\partial DV(t)}{\partial W_i} \right] \\
&= -2E(t)^T \left[\frac{\partial CBV(t-1)}{\partial W_i} + \frac{\partial DV(t)}{\partial W_i} \right] \\
&= -2E(t)^T \left[\frac{\partial CBf\left(U, t-1\right)}{\partial W_i} + \frac{\partial Df\left(U, t\right)}{\partial W_i} \right]
\end{aligned}
$$

$$= -2E(t)^T \frac{\partial}{\partial W_i} \left(\begin{bmatrix} c_{11} & \cdots & c_{1n} \\ c_{21} & \cdots & c_{2n} \\ \vdots & \ddots & \vdots \\ c_{R1} & \cdots & c_{Rn} \end{bmatrix} \begin{bmatrix} b_{11} & \cdots & b_{1P} \\ b_{21} & \cdots & b_{2P} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nP} \end{bmatrix} \begin{bmatrix} v_1(t-1) \\ v_2(t-1) \\ \vdots \\ v_P(t-1) \end{bmatrix} + \begin{bmatrix} d_{11} & \cdots & d_{1P} \\ d_{21} & \cdots & d_{2P} \\ \vdots & \ddots & \vdots \\ d_{R1} & \cdots & d_{RP} \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \\ \vdots \\ v_P(t) \end{bmatrix} \right)$$

$$= -2E(t)^T \left(\begin{bmatrix} cb_{11} & \cdots & cb_{1P} \\ cb_{21} & \cdots & cb_{2P} \\ \vdots & \ddots & \vdots \\ cb_{R1} & \cdots & cb_{RP} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial W_i} v_1(t-1) \\ \frac{\partial}{\partial W_i} v_2(t-1) \\ \vdots \\ \frac{\partial}{\partial W_i} v_P(t-1) \end{bmatrix} + \begin{bmatrix} d_{11} & \cdots & d_{1P} \\ d_{21} & \cdots & d_{2P} \\ \vdots & \ddots & \vdots \\ d_{R1} & \cdots & d_{RP} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial W_i} v_1(t) \\ \frac{\partial}{\partial W_i} v_2(t) \\ \vdots \\ \frac{\partial}{\partial W_i} v_P(t) \end{bmatrix} \right)$$

$$= -2E(t)^T \left(\begin{bmatrix} cb_{11} & \cdots & cb_{1P} \\ cb_{21} & \cdots & cb_{2P} \\ \vdots & \ddots & \vdots \\ cb_{R1} & \cdots & cb_{RP} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial W_i} W_1 \Phi_1(t-1) \\ \frac{\partial}{\partial W_i} W_2 \Phi_2(t-1) \\ \vdots \\ \frac{\partial}{\partial W_i} W_P \Phi_P(t-1) \end{bmatrix} + \begin{bmatrix} d_{11} & \cdots & d_{1P} \\ d_{21} & \cdots & d_{2P} \\ \vdots & \ddots & \vdots \\ d_{R1} & \cdots & d_{RP} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial W_i} W_1 \Phi_1(t) \\ \frac{\partial}{\partial W_i} W_2 \Phi_2(t) \\ \vdots \\ \frac{\partial}{\partial W_i} W_P \Phi_P(t) \end{bmatrix} \right)$$

$$= -2E(t)^T \left(\begin{bmatrix} cb_{1i}\Phi_i(t-1) \\ cb_{2i}\Phi_i(t-1) \\ \vdots \\ cb_{Ri}\Phi_i(t-1) \end{bmatrix} + \begin{bmatrix} d_{1i}\Phi_i(t) \\ d_{2i}\Phi_i(t) \\ \vdots \\ d_{Ri}\Phi_i(t) \end{bmatrix} \right)$$

$$= -2[e_1(t) \quad e_2(t) \quad \cdots \quad e_R(t)] \begin{bmatrix} cb_{1i}\Phi_i(t-1) + d_{1i}\Phi_i(t) \\ cb_{2i}\Phi_i(t-1) + d_{2i}\Phi_i(t) \\ \vdots \\ cb_{Ri}\Phi_i(t-1) + d_{Ri}\Phi_i(t) \end{bmatrix}$$

$$= -2\left[e_1(t)\{cb_{1i}\Phi_i(t-1) + d_{1i}\Phi_i(t)\} \quad + \quad \cdots \quad + \quad e_R(t)\{cb_{Ri}\Phi_i(t-1) + d_{Ri}\Phi_i(t)\}\right]$$

$$\Rightarrow \frac{\partial I}{\partial W_i} = -2\sum_{j=1}^{R} e_j(t)\{cb_{ji}\Phi_i(t-1) + d_{ji}\Phi_i(t)\}$$

$$W_i(k+1) = W_i(k) + 2\eta\sum_{j=1}^{R} e_j(t)\{cb_{ji}\Phi_i(t-1) + d_{ji}\Phi_i(t)\} \tag{4.7}$$

where $e_j(t)$ corresponds to error at the j^{th} output at discrete time instant t, η is the learning rate of the RBFNN, W_i is the vector for the weights of i^{th} nonlinearity, $\Phi_i(t-1)$ and $\Phi_i(t)$ are vectors for basis function of i^{th} nonlinearity at discrete time instants (*t-1*) and t respectively.

$$W_i = [w_1 \quad w_2 \quad w_3 \quad \cdots \quad w_Q]$$

$$\Phi_i(t-1) = [\phi\|u_i(t-1) - c_1\| \quad \phi\|u_i(t-1) - c_2\| \quad \cdots \quad \phi\|u_i(t-1) - c_Q\|]^T$$

$$\Phi_i(t) = [\phi\|u_i(t) - c_1\| \quad \phi\|u_i(t) - c_2\| \quad \cdots \quad \phi\|u_i(t) - c_Q\|]^T$$

4.1.2 Training Algorithm using PSO and Subspace Identification

Like the SISO counterpart, MIMO hammerstein system with separate nonlinearities is also identified using PSO based algorithm. Same steps are followed as those mentioned for the SISO case, with the only difference of updating P sets of weights for P separate RBF networks, each one estimating a separate nonlinearity.

4.2 MIMO System with Combined Nonlinearities

Consider a hammerstein model with P inputs and R outputs. However, instead of P separate nonlinearities, consider a system with a single combined nonlinearity. Figure 4.2 shows a general hammerstein model with combined nonlinearity.

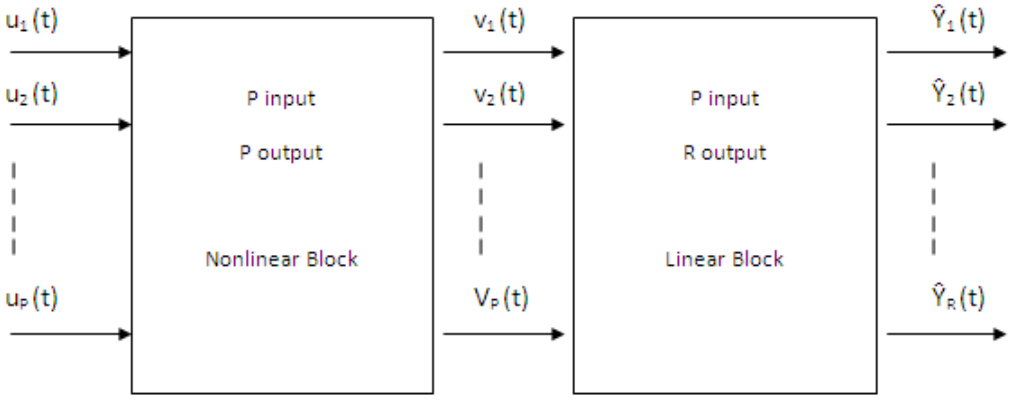

Figure 4.2: A P-input R-output Hammerstein Model with combined nonlinearity

As mentioned earlier, the static nonlinearity is modeled by an RBFNN, but unlike the previous case where a single input was fed to a single RBF network, in this case a complete vector of inputs is fed to a single RBF network at each time instant. Similarly every output node is connected to the same RBF network through a set of linear weights.

Therefore a set of weights W_i for every output v_i of the RBF network is updated. Figure 4.3 shows an RBFNN with P inputs and P outputs.

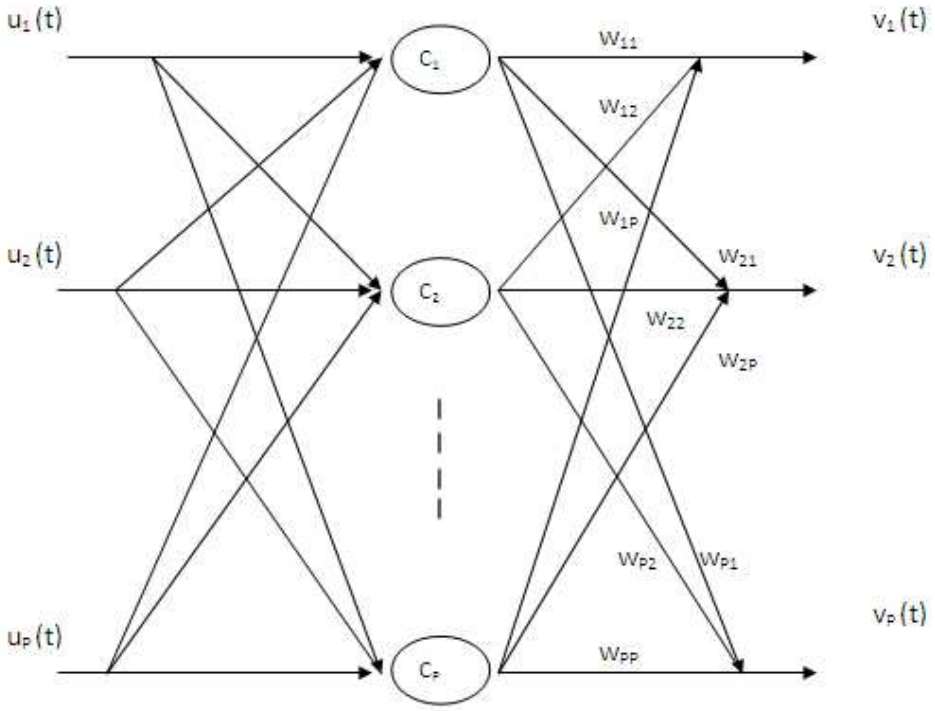

Figure 4.3: A P-input P-output RBFNN

4.2.1 Training Algorithm using LMS principle and Subspace Identification

Using LMS principle like before, the update equation is given by

$$W_i(k+1) = W_i(k) - \eta \frac{\partial I}{\partial W_i} \tag{4.8}$$

where weights W_i correspond to the i^{th} output v_i of the neural network. Cost index is given by

$$I = E(t)^T E(t) \tag{4.9}$$

As the linear block has P inputs and R outputs, the dimensions of the state space matrices will be $A \epsilon \Re^{n \times n}$, $B \epsilon \Re^{n \times P}$, $C \epsilon \Re^{R \times n}$, and $D \epsilon \Re^{R \times P}$. Now finding

the partial derivative of eq 4.9 $w.r.t$ any weights W_i

$$
\begin{aligned}
\frac{\partial I}{\partial W_i} &= \frac{\partial}{\partial W_i} E(t)^T E(t) \\
&= 2E(t)^T \frac{\partial}{\partial W_i} E(t) \\
&= 2E(t)^T \frac{\partial}{\partial W_i} \left[(y_1(t) - \hat{y}_1(t)) \quad (y_2(t) - \hat{y}_2(t)) \quad \cdots \quad (y_R(t) - \hat{y}_R(t)) \right]^T \\
&= 2E(t)^T \frac{\partial}{\partial W_i} \left[-\hat{y}_1(t) \quad -\hat{y}_2(t) \quad \cdots \quad -\hat{y}_R(t) \right]^T \\
&= -2E(t)^T \frac{\partial}{\partial W_i} \left[Cx(t) + DV(t) \right] \\
&= -2E(t)^T \frac{\partial}{\partial W_i} \left[C\{Ax(t-1) + BV(t-1)\} + DV(t) \right] \\
&= -2E(t)^T \left[\frac{\partial CAx(t-1)}{\partial W_i} + \frac{\partial CBV(t-1)}{\partial W_i} + \frac{\partial DV(t)}{\partial W_i} \right] \\
&= -2E(t)^T \left[\frac{\partial CBV(t-1)}{\partial W_i} + \frac{\partial DV(t)}{\partial W_i} \right] \\
&= -2E(t)^T \left[\frac{\partial CBf\left(U, t-1\right)}{\partial W} + \frac{\partial Df\left(U, t\right)}{\partial W} \right]
\end{aligned}
$$

$$= -2E(t)^T \frac{\partial}{\partial W_i} \left(\left[\begin{array}{ccc} c_{11} & \cdots & c_{1n} \\ c_{21} & \cdots & c_{2n} \\ \vdots & \ddots & \vdots \\ c_{R1} & \cdots & c_{Rn} \end{array} \right] \left[\begin{array}{ccc} b_{11} & \cdots & b_{1M} \\ b_{21} & \cdots & b_{2M} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nM} \end{array} \right] \left[\begin{array}{c} v_1(t-1) \\ v_2(t-1) \\ \vdots \\ v_M(t-1) \end{array} \right] + \left[\begin{array}{ccc} d_{11} & \cdots & d_{1M} \\ d_{21} & \cdots & d_{2M} \\ \vdots & \ddots & \vdots \\ d_{R1} & \cdots & d_{RM} \end{array} \right] \left[\begin{array}{c} v_1(t) \\ v_2(t) \\ \vdots \\ v_M(t) \end{array} \right] \right)$$

$$= -2E(t)^T \left(\left[\begin{array}{ccc} cb_{11} & \cdots & cb_{1M} \\ cb_{21} & \cdots & cb_{2M} \\ \vdots & \ddots & \vdots \\ cb_{R1} & \cdots & cb_{RM} \end{array} \right] \left[\begin{array}{c} \frac{\partial}{\partial W_i} v_1(t-1) \\ \frac{\partial}{\partial W_i} v_2(t-1) \\ \vdots \\ \frac{\partial}{\partial W_i} v_M(t-1) \end{array} \right] + \left[\begin{array}{ccc} d_{11} & \cdots & d_{1M} \\ d_{21} & \cdots & d_{2M} \\ \vdots & \ddots & \vdots \\ d_{R1} & \cdots & d_{RM} \end{array} \right] \left[\begin{array}{c} \frac{\partial}{\partial W_i} v_1(t) \\ \frac{\partial}{\partial W_i} v_2(t) \\ \vdots \\ \frac{\partial}{\partial W_i} v_M(t) \end{array} \right] \right)$$

$$= -2E(t)^T \left(\begin{bmatrix} cb_{11} & \cdots & cb_{1M} \\ cb_{21} & \cdots & cb_{2M} \\ \vdots & \ddots & \vdots \\ cb_{R1} & \cdots & cb_{RM} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial W_i} W_1 \Phi(t-1) \\ \frac{\partial}{\partial W_i} W_2 \Phi(t-1) \\ \vdots \\ \frac{\partial}{\partial W_i} W_M \Phi(t-1) \end{bmatrix} + \begin{bmatrix} d_{11} & \cdots & d_{1M} \\ d_{21} & \cdots & d_{2M} \\ \vdots & \ddots & \vdots \\ d_{R1} & \cdots & d_{RM} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial W_i} W_1 \Phi(t) \\ \frac{\partial}{\partial W_i} W_2 \Phi(t) \\ \vdots \\ \frac{\partial}{\partial W_i} W_M \Phi(t) \end{bmatrix} \right)$$

$$= -2E(t)^T \left(\begin{bmatrix} cb_{1i} \Phi(t-1) \\ cb_{2i} \Phi(t-1) \\ \vdots \\ cb_{Ri} \Phi(t-1) \end{bmatrix} + \begin{bmatrix} d_{1i} \Phi(t) \\ d_{2i} \Phi(t) \\ \vdots \\ d_{Ri} \Phi(t) \end{bmatrix} \right)$$

$$= -2 \begin{bmatrix} e_1(t) & e_2(t) & \cdots & e_R(t) \end{bmatrix} \begin{bmatrix} cb_{1i} \Phi(t-1) + d_{1i} \Phi(t) \\ cb_{2i} \Phi(t-1) + d_{2i} \Phi(t) \\ \vdots \\ cb_{Ri} \Phi(t-1) + d_{Ri} \Phi(t) \end{bmatrix}$$

$$= -2 \left[e_1(t) \{ cb_{1i} \Phi(t-1) + d_{1i} \Phi(t) \} \quad + \quad \cdots \quad + \quad e_R(t) \{ cb_{Ri} \Phi(t-1) + d_{Ri} \Phi(t) \} \right]$$

$$\Rightarrow \frac{\partial I}{\partial W_i} = -2\sum_{j=1}^{R} e_j(t)\{cb_{ji}\Phi(t-1) + d_{ji}\Phi(t)\}$$

$$W_i(k+1) = W_i(k) + 2\eta\sum_{j=1}^{R} e_j(t)\{cb_{ji}\Phi(t-1) + d_{ji}\Phi(t)\} \tag{4.10}$$

where $e_j(t)$ corresponds to error at the j^{th} output at discrete time instant t, η is the learning rate of the RBFNN, $\Phi(t-1)$ and $\Phi(t)$ are the vectors for the basis function at discrete time *(t-1)* and t, and W_i is the vector for the weights corresponding to the i^{th} nonlinear output $v_i(t)$.

4.2.2 Training Algorithm using PSO and Subspace Identification

PSO based algorithm is used to estimate MIMO hammerstein system with combined nonlinearity. Again, the same steps are followed as mentioned for the SISO case, with the difference that P different sets of weights are updated. Unlike the separate nonlinearities' case, these P sets of weights are sets for a single RBF network, and not for P separate networks. Each set of weights W_i represents weights for i^{th} output of the neural network.

CHAPTER 5

SIMULATION RESULTS

5.1 Results for SISO system

5.1.1 Example 1: Square root nonlinearity

The first example considers the following hammerstein type nonlinear process whose static nonlinearity is given by

$$v(t) = sign\,(u(t))\,\sqrt{|u(t)|}$$

The dynamic linear part is given by a third order discrete time state-space system

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \end{bmatrix} = \begin{bmatrix} 1.80 & 1 & 0 \\ -1.07 & 0 & 1 \\ 0.21 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 4.80 \\ 1.93 \\ 1.21 \end{bmatrix} v(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$$

The linear dynamic part of the system has eigen values at

$$\lambda_1 = 0.7, \quad \lambda_2 = 0.6 \quad and \quad \lambda_3 = 0.5$$

The identification structure comprises of RBFNN cascaded with a state-space system given by

$$x(t+1) = Ax(t) + Bv(t)$$

$$\hat{y(t)} = Cx(t) + Dv(t)$$

Results using LMS based algorithm

The original system is excited using uniformly distributed random numbers in the interval $[-1.75, 1.75]$ to generate the data. The centers for the RBFNN are distributed evenly in the above mentioned interval. The famous *k-means* method is employed to generate the centers of the RBFNN. The spread of the basis function and learning rate of the neurons is selected to have an optimum value of 0.65 and 10^{-5} respectively after hit and trial.

The proposed algorithm employing LMS principle and subspace identification is used to identify the system. With an RBF network of 10 neurons, the mean

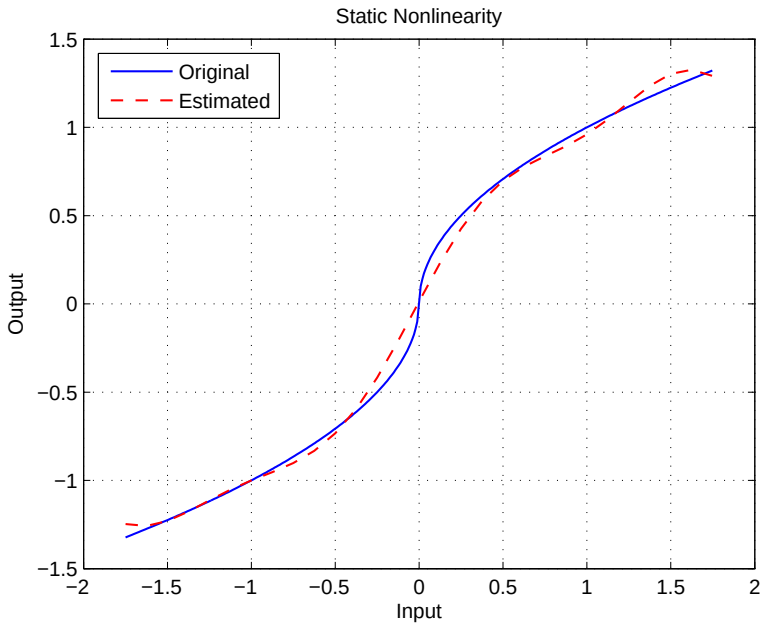

Figure 5.1: Square root nonlinearity for hammerstein model of example 1 identified by LMS based algorithm

squared error between normalized values of actual and estimated outputs converges to 2×10^{-4} in just 25 iterations. The comparison of the identified and actual nonlinearity is shown in figure 5.1. The linear dynamic part is also identified with accuracy. Eigen values of the estimated linear part of the system are

$$\lambda_1 = 0.7173, \quad \lambda_2 = 0.5428 + 0.0539i \quad and \quad \lambda_3 = 0.5428 - 0.0539i$$

which are very close to the original eigen values. Step response of the linear dynamic part is shown in figure 5.2. The mean squared error plot is shown in figure 5.3.

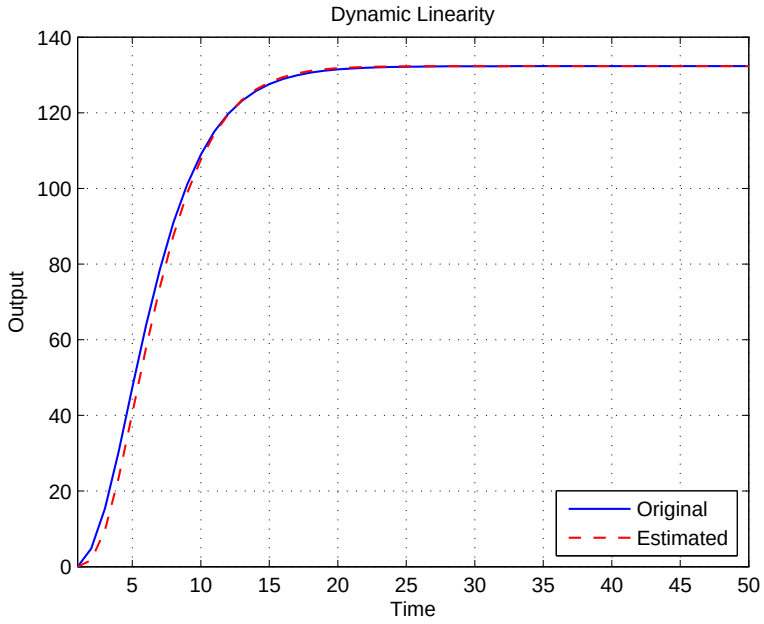

Figure 5.2: Step response of linear dynamic part of hammerstein model of example 1 identified by LMS based algorithm

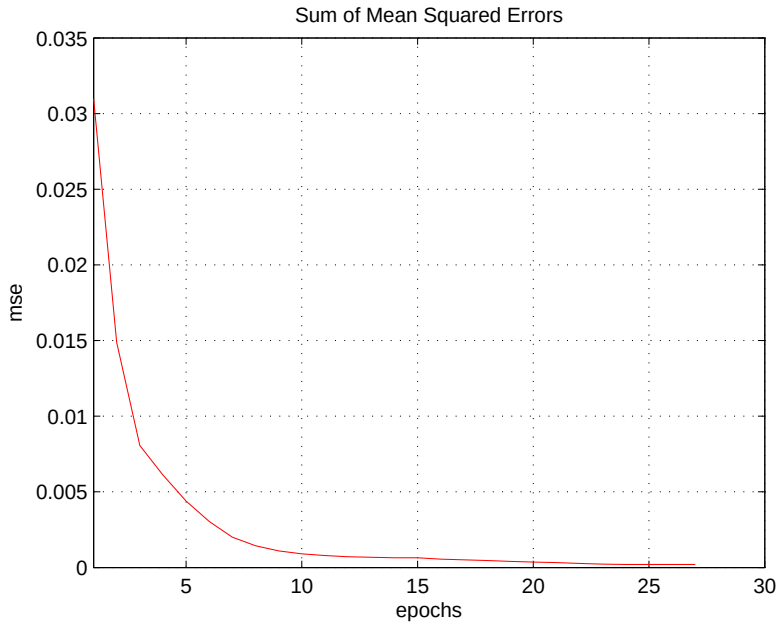

Figure 5.3: Squared error for hammerstein model of example 1 identified by LMS based algorithm

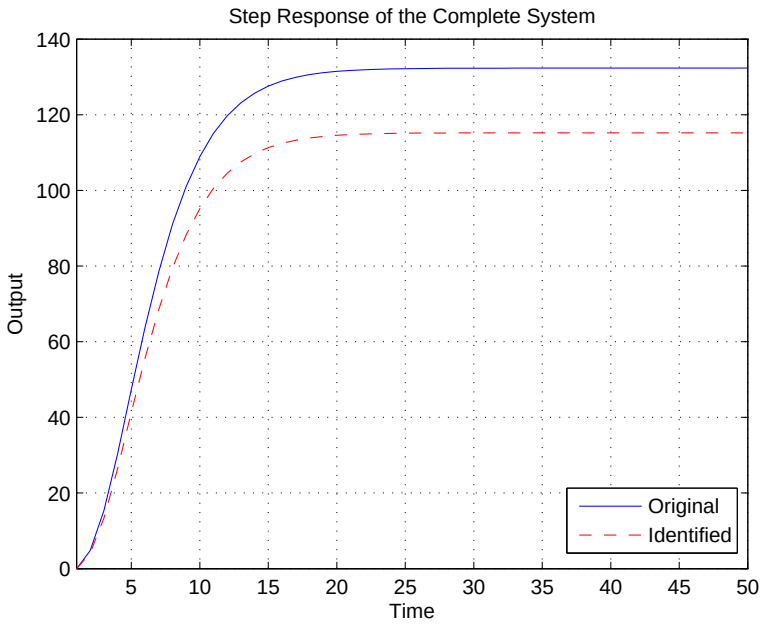

Figure 5.4: Step response of complete hammerstein system of example 1 identified by LMS based algorithm

The magnitude of response of the estimated nonlinearity is shifted from the magnitude of response of the actual nonlinearity by a factor. The same is true for the response of the dynamic linear part. But naturally, the factor by which the magnitude of the nonlinear response is shifted is reciprocal of the factor by which the linear response is shifted. Hence, the total response of the estimated system is the same as that of the original system.

This idea is verified by the fact that when we excite the original and estimated systems with a step input, output responses are very close. Figure 5.4 shows the step response of the complete systems. It should be noted that this step response is not to be confused with the step response shown in figure 5.2 which is the step response of only the linear dynamic part and not of the complete system.

Results using PSO based algorithm

The second algorithm employing PSO and subspace identification is then used to identify the same system. The same generated data is used. An RBF network of 10 neurons is initialized with centers evenly distributed in the interval $[-1.75, 1.75]$. Within 13 iterations the mean squared error between the normalized values of original and estimated outputs converges to 4×10^{-4}. Figures 5.5 and 5.6 show the estimation of static noninearity and dynamic linearity respectively. Figure 5.7 shows the convergence of the mean squared error. The eigen values of the estimated linear part are

$$\lambda_1 = 0.6811, \quad \lambda_2 = 0.6481 \quad \textit{and} \quad \lambda_3 = 0.4639$$

which are again quite close to the eigen values of the original system.

Comparison of results

Both algorithms have estimated the nonlinearity with fairly good accuracy. The output mean squared error has converged to a much smaller value when estimation using LMS based algorithm is carried out. Moreover LMS algorithm has taken much lesser time than PSO based identification, as each iteration of the former algorithm takes much less time than each iteration of the later.

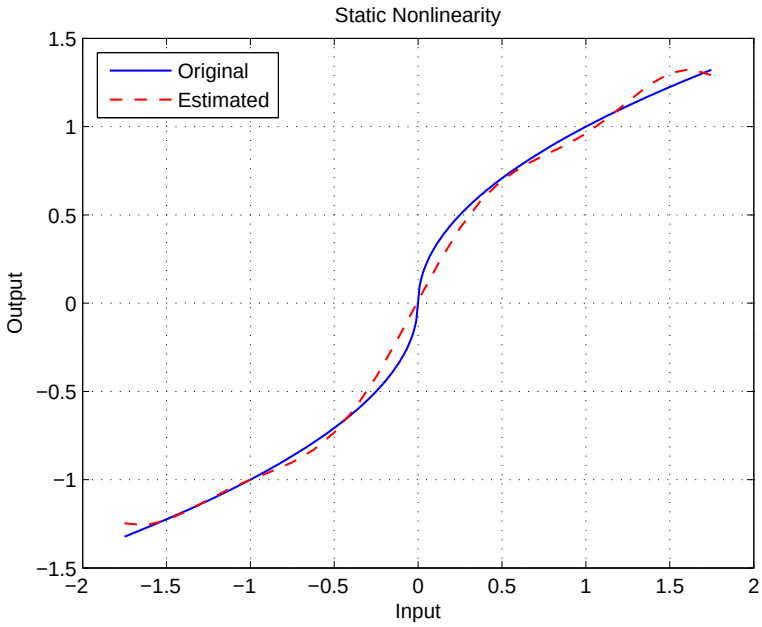

Figure 5.5: Square root nonlinearity for hammerstein model of example 1 identified by PSO based algorithm

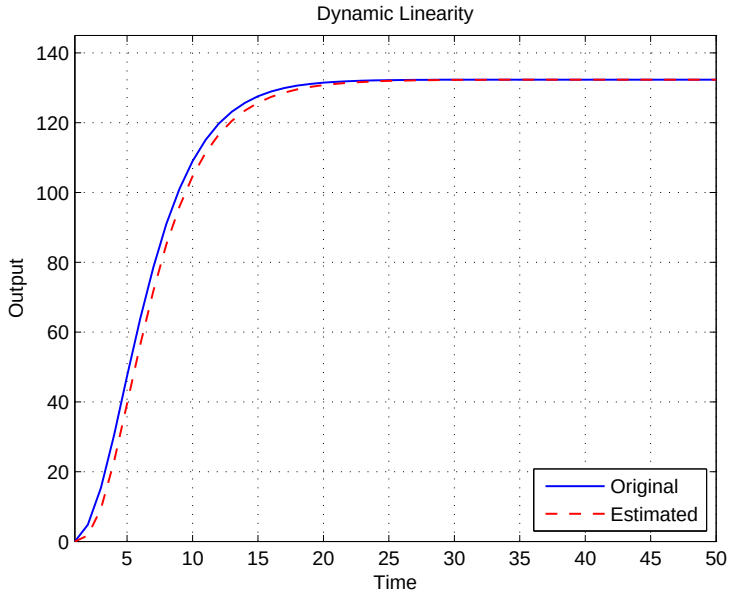

Figure 5.6: Step response of linear dynamic part of hammerstein model of example 1 identified by PSO based algorithm

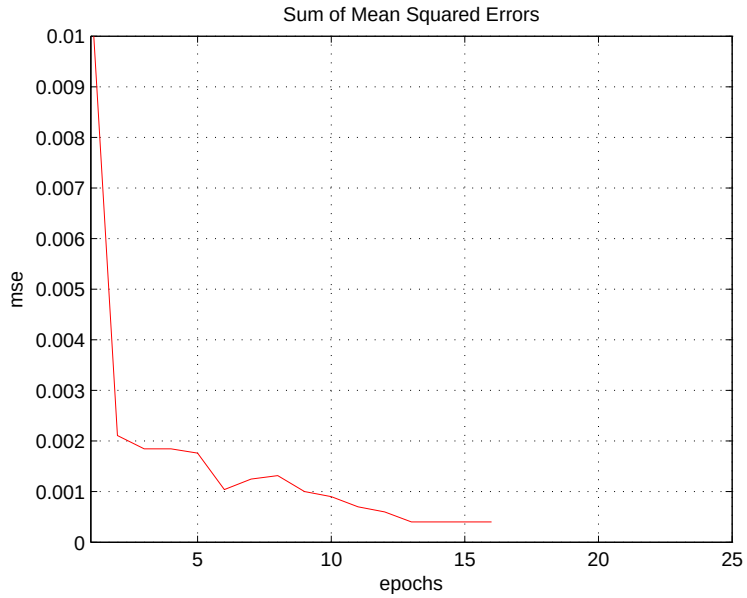

Figure 5.7: Squared error for hammerstein model of example 1 identified by PSO based algorithm

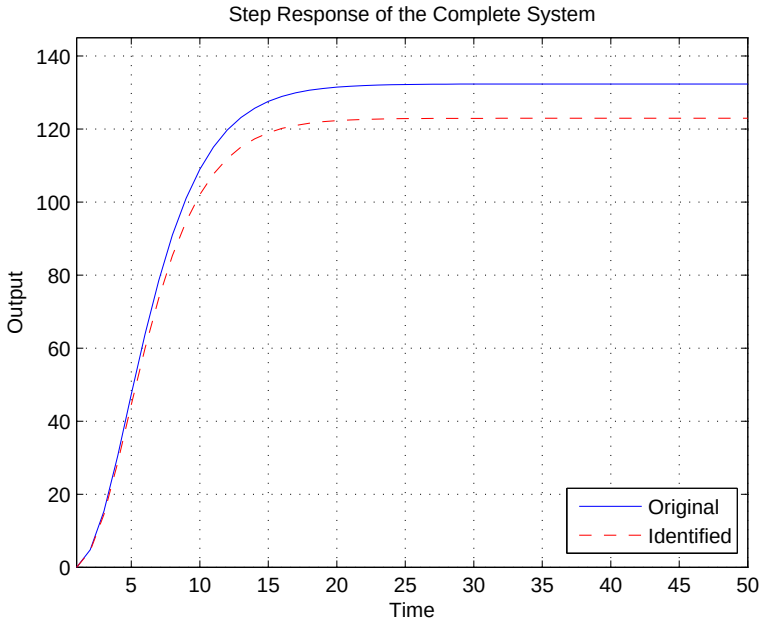

Figure 5.8: Step response of complete hammerstein system of example 1 identified by PSO based algorithm

5.1.2 Example 2: Exponential/tanh nonlinearity

The second example considers the following hammerstein type nonlinear process whose static nonlinearity is much difficult to learn and is given in [82] by

$$\begin{aligned} v(t) &= tanh\,[2u(t)] \qquad 1.5 \geq u(t) \\ v(t) &= \frac{exp(u(t)) - 1}{exp(u(t)) + 1} \qquad 4 > u(t) > 1.5 \end{aligned}$$

The dynamic linear part is given by the following second order discrete time state-space system

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -0.5 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} v(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

The linear part of the system has eigen values at

$$\lambda_1 = 0.5 + 0.5i$$

$$\lambda_2 = 0.5 - 0.5i$$

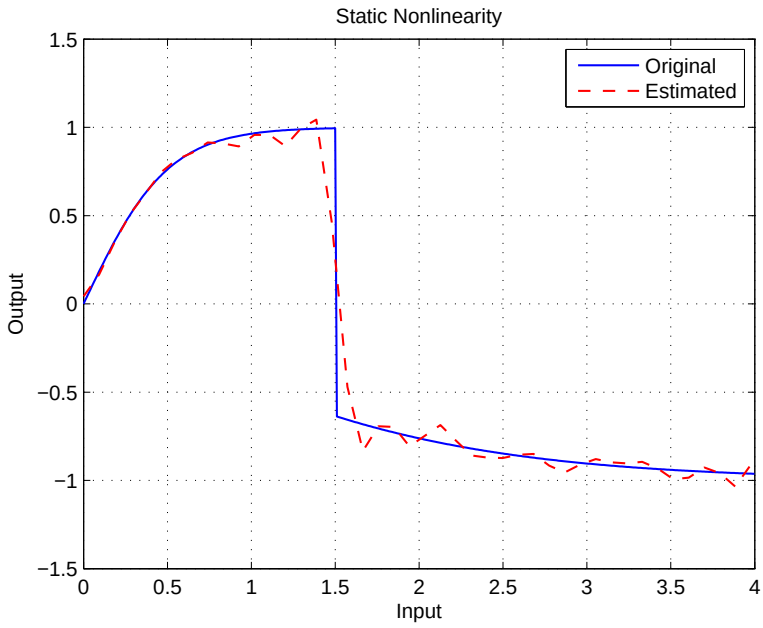

Figure 5.9: Exponential/tanh nonlinearity for hammerstein model of example 2 identified by LMS based algorithm

Results using LMS based algorithm

The system is excited using uniform random numbers in the interval $[0, 4]$ and data is generated. The identification structure is composed of an RBFNN for the nonlinear part. The linear part is modeled by a state space model of second order. Centers of the RBFNN are evenly distributed in the interval $[0, 4]$ and are chosen using the *k-means* method. The spread of the basis function and learning rate of the neurons are selected to have an optimum value of 0.17 and 0.01 respectively after hit and trial.

An RBF network consisting of 35 neurons is used for identification. After 20 iterations, the mean squared error converges to a final value of 8×10^{-4}. The estimates of nonlinearity and linearity are shown respectively in figure 5.9 and

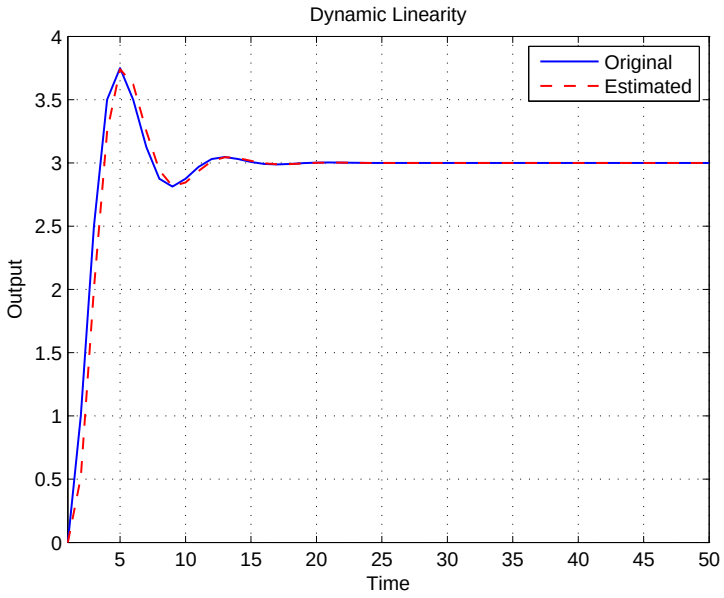

Figure 5.10: Step response of linear dynamic part of hammerstein model of example 2 identified by LMS based algorithm

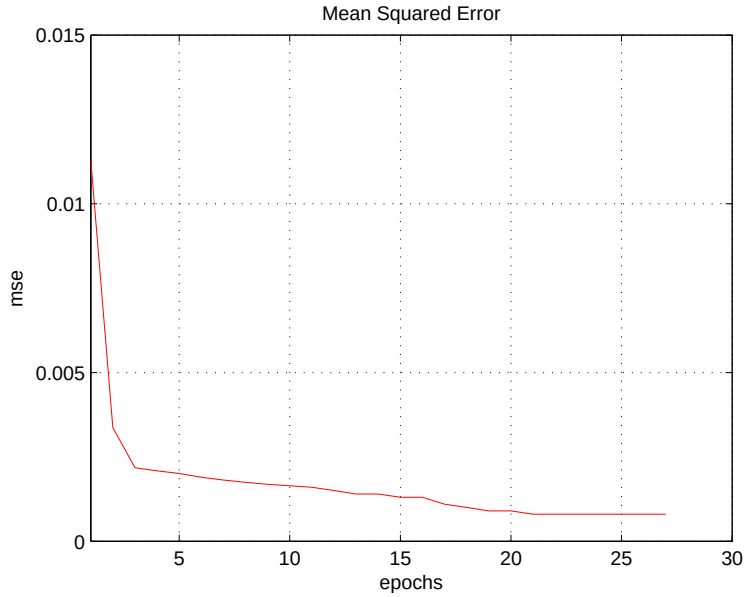

Figure 5.11: Squared error for hammerstein model of example 2 identified by LMS based algorithm

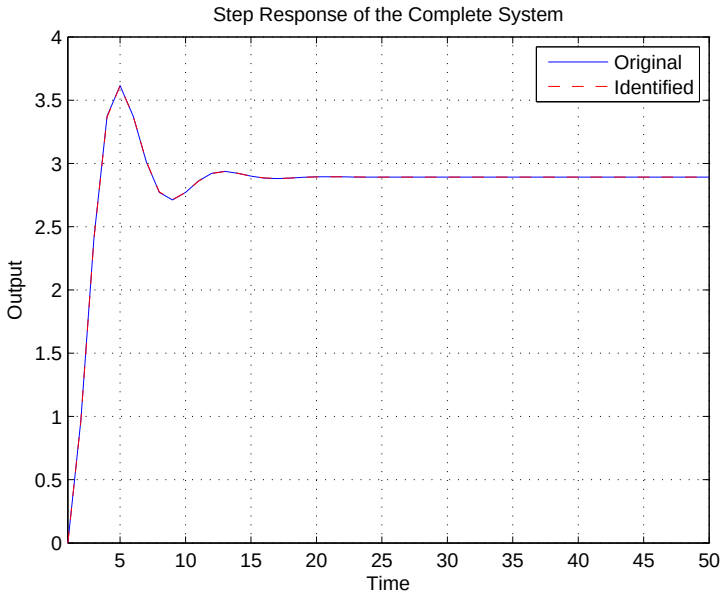

Figure 5.12: Step response of complete hammerstein system of example 2 identified by LMS based algorithm

figure 5.10. The squared error plot is shown in figure 5.11 while figure 5.12 shows the step response of the complete system. The eigen values of the linear part of the estimated system are

$$\lambda_1 = 0.497 + 0.5i$$

$$\lambda_2 = 0.497 - 0.5i$$

Results using PSO based algorithm

The system is then identified using PSO based algorithm. Once again, the RBFNN is initialized with a set of 35 neurons. Same set of centers is used as was used in the LMS case. The mean squared error converges to 1.6×10^{-3} after 25 iterations . Figures 5.13 and 5.14 show the estimates of noninearity and linearity respectively.

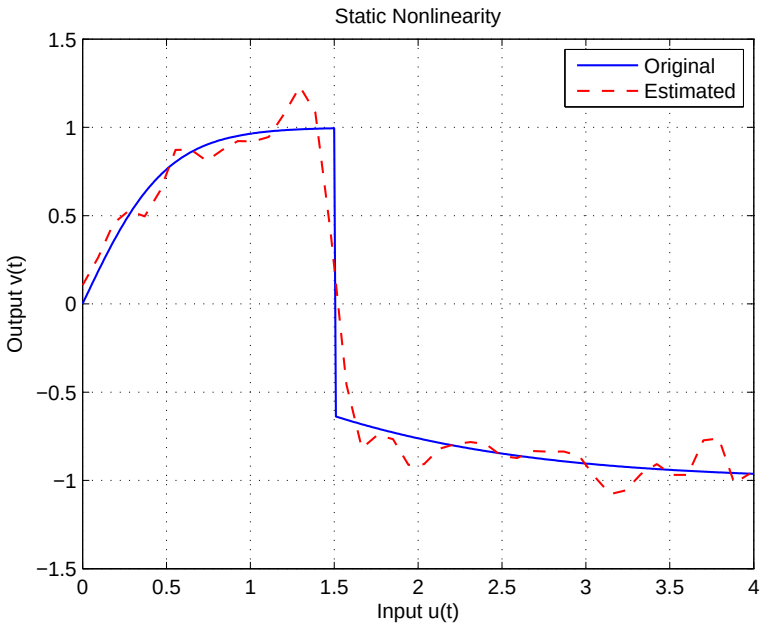

Figure 5.13: Exponential/tanh nonlinearity for hammerstein model of example 2 identified by PSO based algorithm

Figure 5.15 shows the convergence of the squared error. The eigen values of the linear part of the estimated system are

$$\lambda_1 = 0.5010 + 0.5035i$$

$$\lambda_2 = 0.5010 - 0.5035i$$

which are approximately the same eigen values as those of the original linear part.

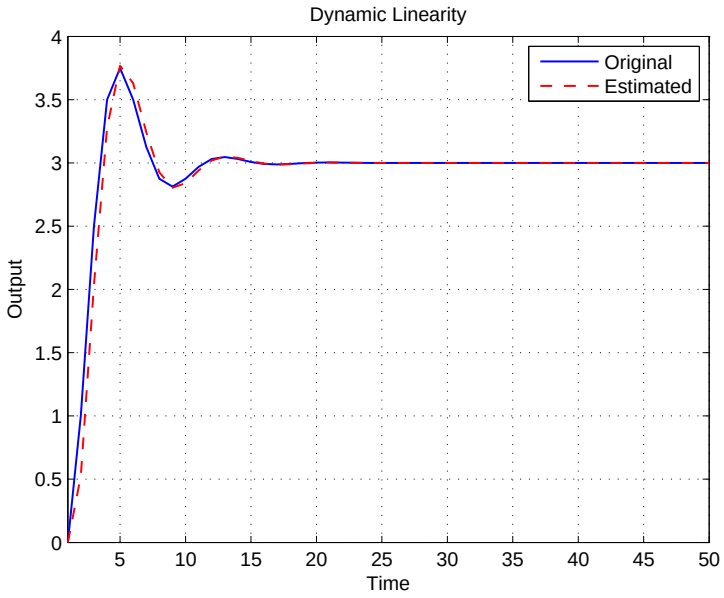

Figure 5.14: Step response of linear dynamic part of hammerstein model of example 2 identified by PSO based algorithm

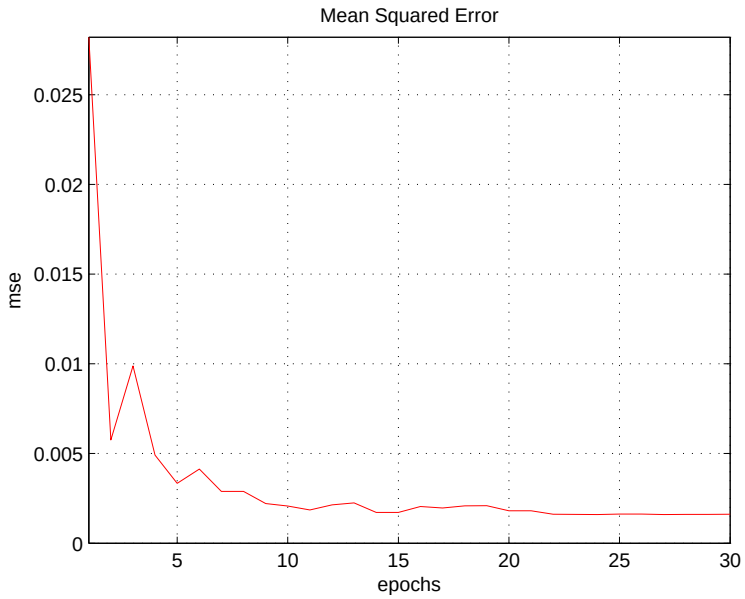

Figure 5.15: Squared error for hammerstein model of example 2 identified by PSO based algorithm

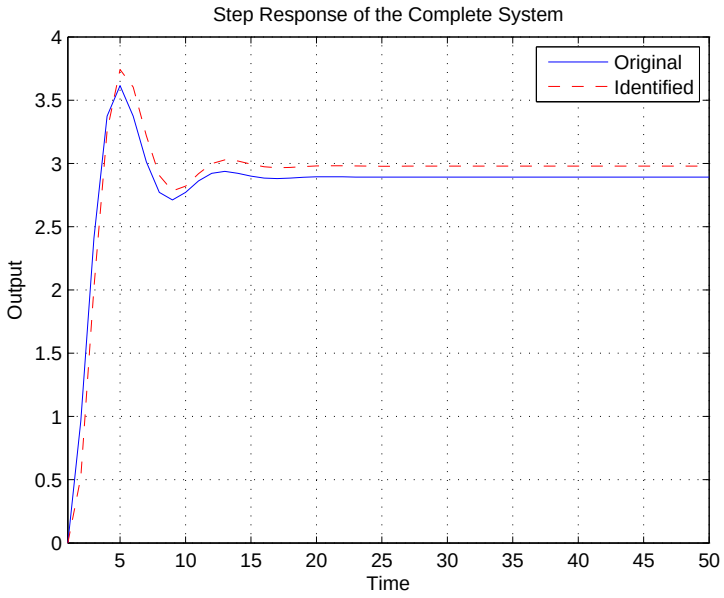

Figure 5.16: Step response of complete hammerstein system of example 2 identified by PSO based algorithm

Comparison of results

Both LMS based and PSO based algorithms have estimated a good approximation of the original system. The nonlinearity is estimated with considerably good accuracy in both the cases. Mean squared error converges to a smaller final value in the LMS case. Both algorithms converge to a final state in almost the same number of iterations. However it must be kept in mind that each iteration of PSO takes much longer time than that of the LMS based algorithm. Therefore, in terms of time, LMS based algorithm estimates the system much quicker than PSO based detection. With respect to estimation also, the LMS algorithm gives a slightly better estimate.

Initially an RBF network of 10 neurons was tried. This however did not estimate the shape of the nonlinearity very accurately. Number of neurons was

then increased to 25, and then 35, which has finally given satisfactory results. The achieved results for estimation of nonlinearity are much better than those obtained in [82].

5.2 Results for MIMO system

5.2.1 Example 1: Exponential/tanh and Saturation Nonlinearities

This example considers a 2 input 2 output hammerstein type nonlinear process with two separate static nonlinearities. The first nonlinearity is a saturation nonlinearity while the second nonlinearity is the same as that of example 2 *i.e.* tanh and exponential function

$$v_1(t) = \begin{cases} -0.5 & \text{if } -0.5 \geq u_1(t) \\ u_1(t) & \text{if } 0.5 > u_1(t) > -0.5 \\ 0.5 & \text{if } u_1(t) > 0.5 \end{cases}$$

The dynamic linear part is given by the following second order discrete time state-space system

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -0.5 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 & 0.2 \\ 0.5 & 1 \end{bmatrix} v(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

The linear part of the system has eigen values at

$$\lambda_1 = 0.5 + 0.5i$$

$$\lambda_2 = 0.5 - 0.5i$$

Results using LMS based algorithm

The system is excited using uniform random numbers in the interval $[-1, 1]$ and $[0, 4]$ for the two inputs respectively and data is generated. An RBF network of 35 neurons is selected to identify each nonlinearity. The centers of the RBFNNs are selected using the *k-means* method as before. The spread of the basis function for exponential/tanh and saturation nonlinearity are both optimized to a value of 0.15. The learning rate of the neural network is selected to have an optimum value of 10^{-4} after hit and trial.

The LMS based algorithm identifies the system very accurately. Mean squared error between the normalized values of original and estimated outputs converges to a final value of 5×10^{-4} in only 15 iterations. The static nonlinearities are learnt with accuracy. The linear part is also identified very closely. Estimates of nonlinearities and step responses of the linear part are shown respectively in figures 5.17, 5.18, 5.19 and 5.20. The squared output error plot is shown in figure 5.21, while figure 5.22 shows the step response of the complete identified and

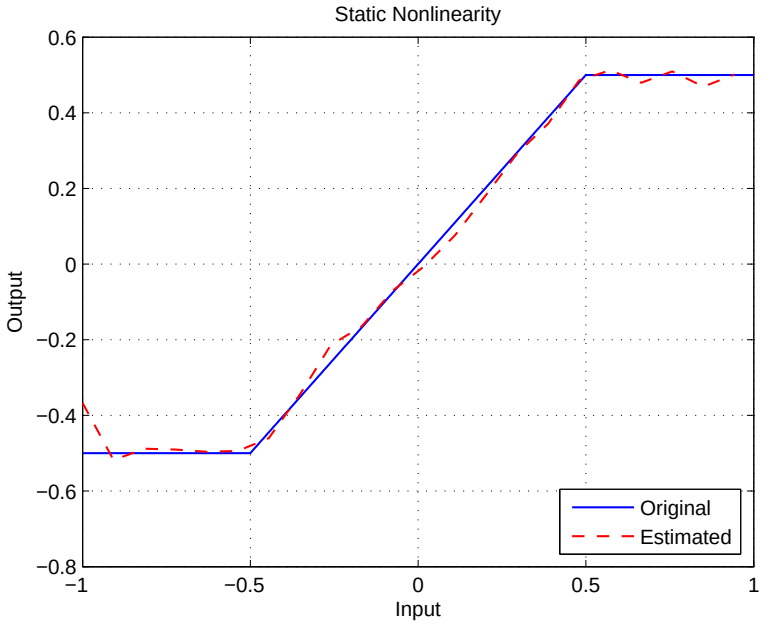

Figure 5.17: Saturation nonlinearity for hammerstein model of MIMO example 1 identified by LMS based algorithm

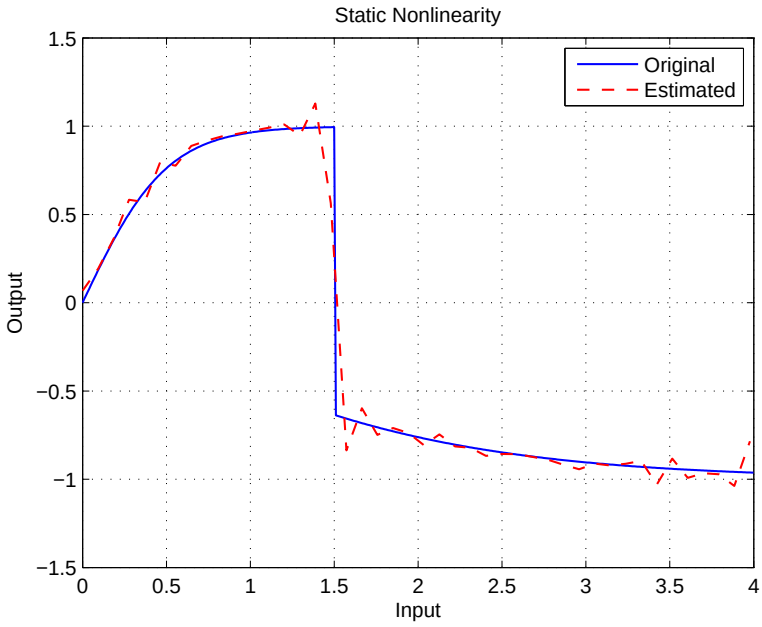

Figure 5.18: Exponential/tanh nonlinearity for hammerstein model of MIMO example 1 identified by LMS based algorithm

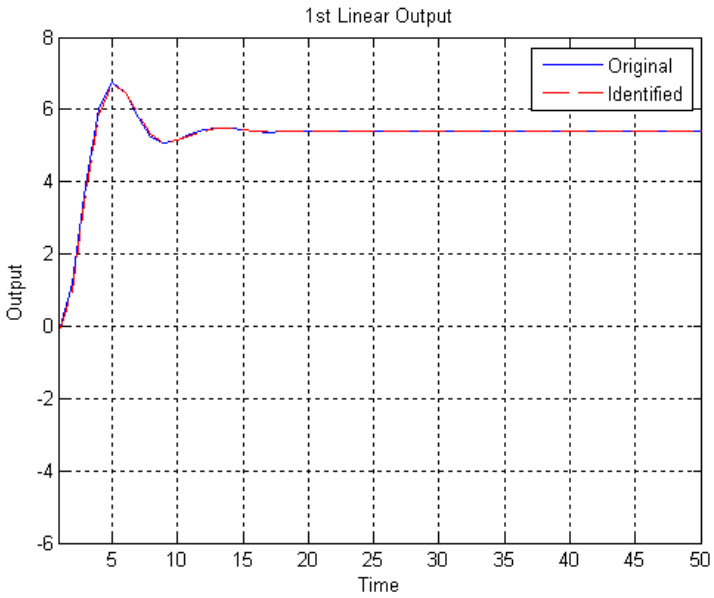

Figure 5.19: Step response (output y_1) of linear dynamic part of hammerstein model of MIMO example 1 identified by LMS based algorithm

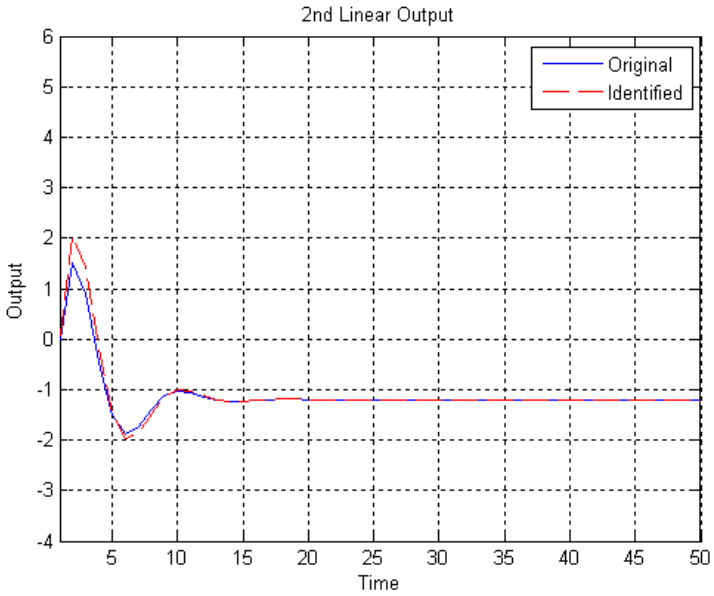

Figure 5.20: Step response (output y_2) of linear dynamic part of hammerstein model of MIMO example 1 identified by LMS based algorithm

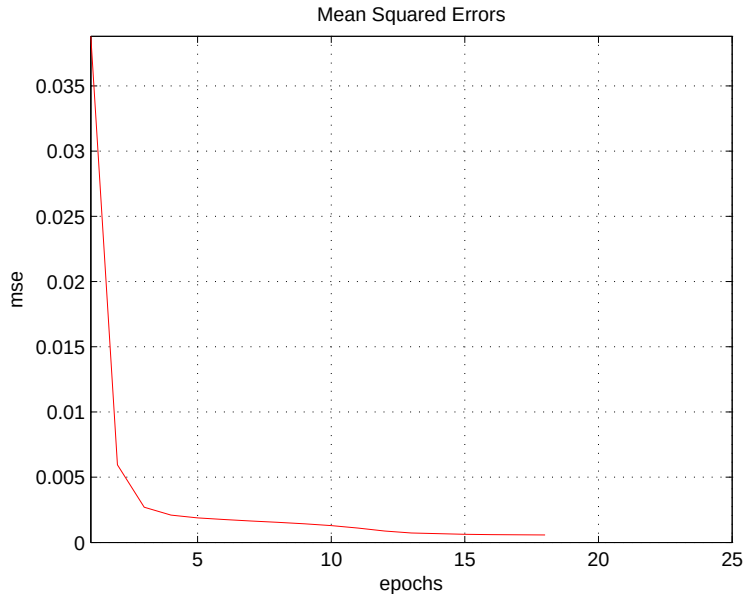

Figure 5.21: Squared output error for hammerstein model of MIMO example 1 identified by LMS based algorithm

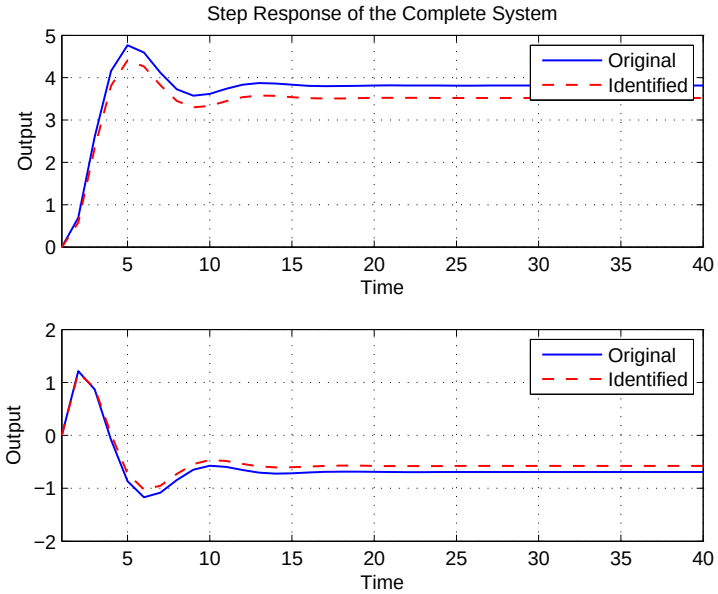

Figure 5.22: Step response (outputs y_1 and y_2) of complete hammerstein system of MIMO example 1 identified by LMS based algorithm

original systems. The eigen values of the estimated linear part are

$$\lambda_1 = 0.5011 + 0.5010i$$

$$\lambda_2 = 0.5011 - 0.5010i$$

Results using PSO based algorithm

The system is then identified using PSO based algorithm. The same generated data is used. The RBFNNs are initialized with the same number of neurons and same set of evenly distributed centers in their respective intervals. The mean squared error converges to a final value of 9×10^{-4} after 24 iterations . Figures 5.23, 5.24 show the estimates of noninearities. Step responses of linear part can be seen in figures 5.25 and 5.26. Figures 5.27 and 5.28 show the step response of the complete system and the squared output error respectively. The eigen values of the estimated linear part are

$$\lambda_1 = 0.4970 + 0.4993i$$

$$\lambda_2 = 0.4970 - 0.4993i$$

Comparison of results

Both algorithms have identified the system. The LMS algorithm again gives better results. The PSO algorithm converges to a slightly larger final value of mean squared error in almost the same number of iterations. LMS based algorithm

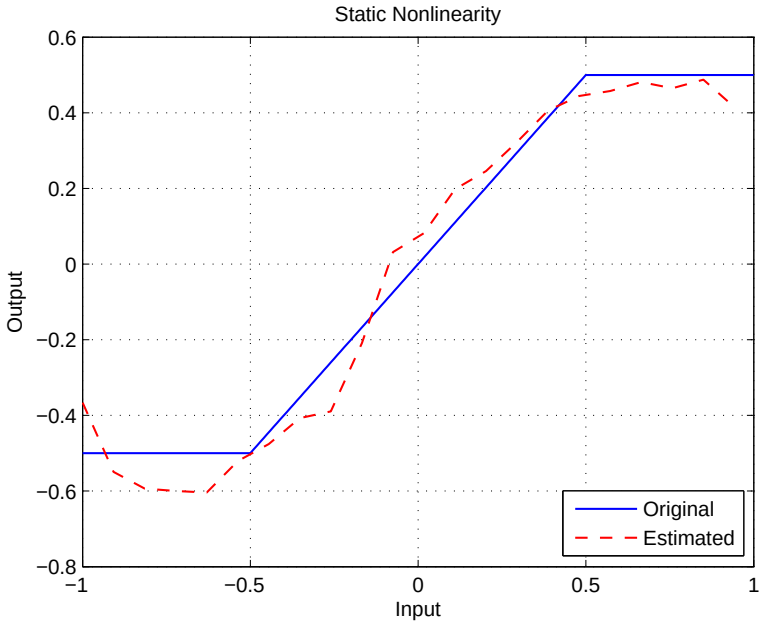

Figure 5.23: Saturation nonlinearity for hammerstein model of MIMO example 1 identified by PSO based algorithm

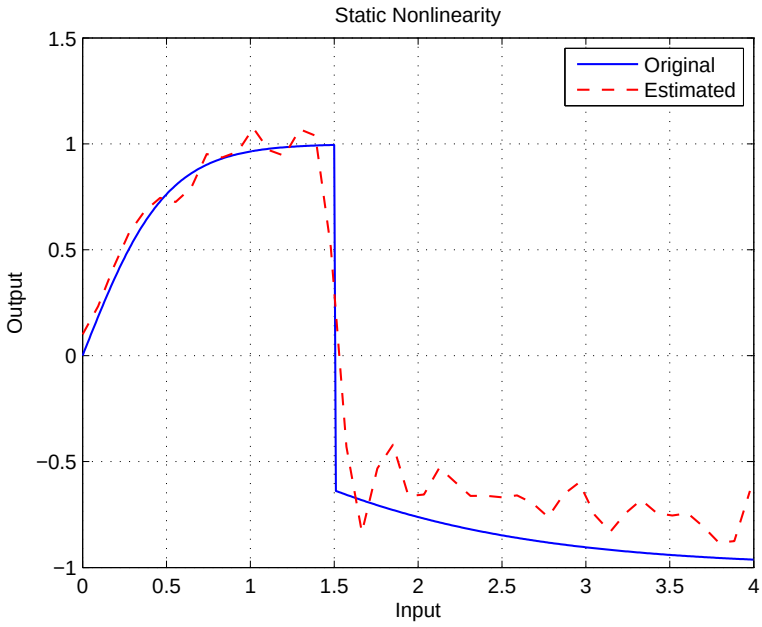

Figure 5.24: Exponential/tanh nonlinearity for hammerstein model of MIMO example 1 identified by PSO based algorithm

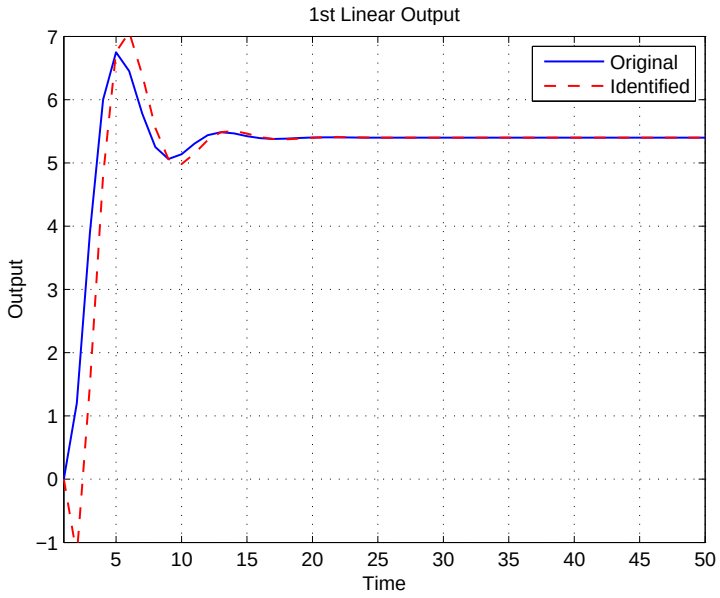

Figure 5.25: Step response (output y_1) of linear dynamic part of hammerstein model of MIMO example 1 identified by PSO based algorithm

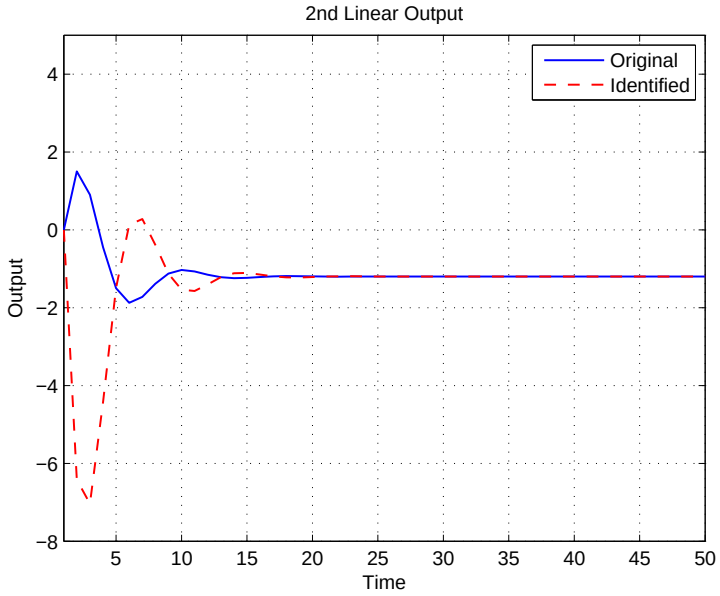

Figure 5.26: Step response (output y_2) of linear dynamic part of hammerstein model of MIMO example 1 identified by PSO based algorithm

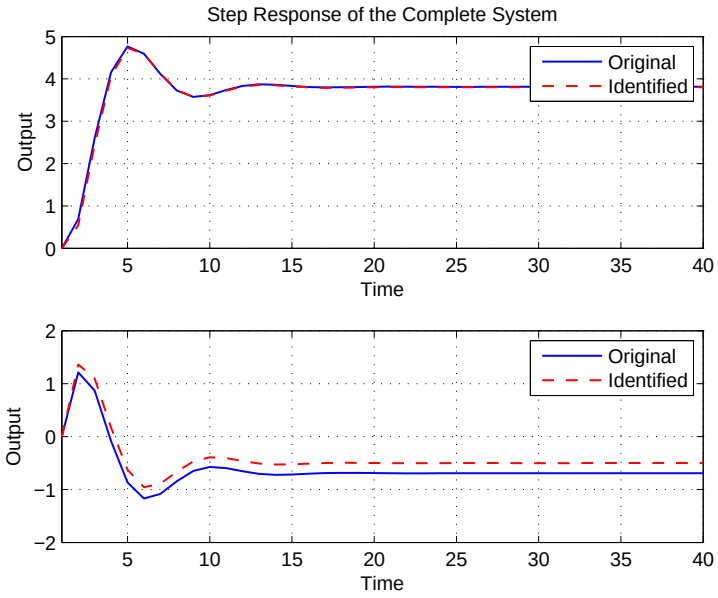

Figure 5.27: Step response (outputs y_1 and y_2) of complete hammerstein system of MIMO example 1 identified by PSO based algorithm

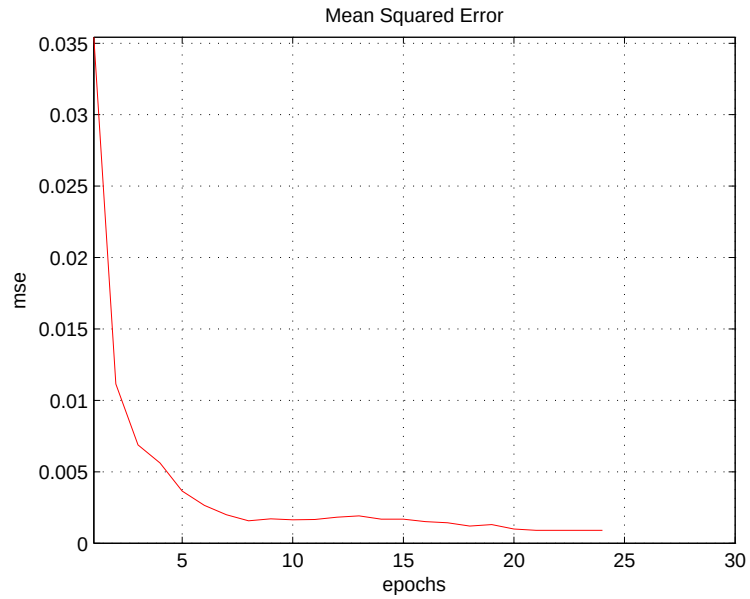

Figure 5.28: Squared error for hammerstein model of MIMO example 1 identified by PSO based algorithm

estimates the shapes of the nonlinearities comparatively better than PSO. Once more, it should be noted that although PSO based algorithm identifies the system in almost the same number of iterations, the time it takes is more than that taken by LMS algorithm.

5.2.2 Example 2: Steam Generator

This example considers a 4 input 4 output steam generator at Abbott Power Plant in Campaign, IL, which is a dual fuel (oil/gas) fired unit used for heating and electric power generation. The plant is rated at 22.096 kg/s of steam at 22.4 MPa (325psi) of pressure. Detailed plant description is given in [83] and [84]. The plant has several boiler dynamics including nonlinearities, instabilities, time delays and load disturbances. A diagram of the steam plant with all its inputs and outputs is shown in figure 5.29.

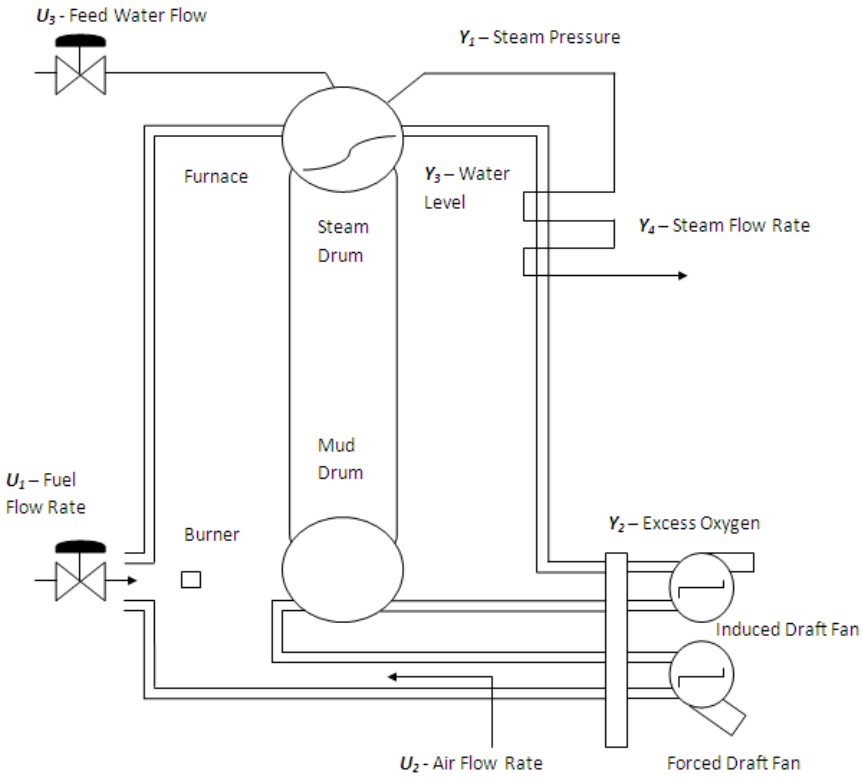

Figure 5.29: Industrial Steam Generation Plant

A description of the plant inputs and outputs is given as,

- Inputs:
  - u_1: Fuel flow rate (scaled 0-1)
  - u_2: Air flow rate (scaled 0-1)
  - u_3: Reference level (inches)
  - u_4: Disturbance defined by the load level (scaled 0-1)
- Outputs:
  - y_1: Drum pressure (psi)
  - y_2: Excess Oxygen in exhaust gases (0-100%)
  - y_3: Level of water in the drum (inches)
  - y_4: Steam Flow (kg/s)

Apart from these measurable outputs and inputs, there are certain disturbances in the plant like *changes in steam demand by users* and *sensor noise*, and certain uncertainties which include *fuel calorific value variations*, *heat transfer coefficient variations*, and *distributed dynamics of steam generation*. The plant also has few constraints like *actuator constraints*, *unidirectional flow rates* and *drum flooding*.

A data set for the plant containing 9600 samples is taken from [85]. Out of these, 5000 samples are used for training, and the last 4600 samples are used for validation. The results of identification using the two proposed algorithms are presented below.

Estimation of plant order

Order estimation for nonlinear and linear systems is a difficult task. To estimate the order of dynamics from input-output data set, several methods have been proposed in the literature. For this example, order is estimated by estimating the rank of the covariance matrix. This method was proposed by Ljung in [86]. The method is described here in brief.

Given a true system defined by,

$$y(t) + a_1 y(t-1) + \cdots + a_n y(t-n) = b_1 u(t-1) + \cdots + b_n u(t-n) + v_0(t)$$

where $v_0(t)$ is some noise sequence, and n is the true order of the system. Then let,

$$\phi_s(t) = [-y(t-1) \cdots - y(t-s) \quad u(t-1) \cdots u(t-s)]^T$$

Supposing $v_0(t)$ is zero, the matrix

$$R^S(N) = \frac{1}{N} \sum_{t=1}^{N} \phi_S(t) \phi_S^T(t)$$

will be nonsingular for $s \leq n$ provided $u(t)$ is persistently exciting, and would be singular for $s \geq n+1$. An eighth order system is estimated for the steam generator by this method. After order estimation, identification is carried out using the two algorithms. The results are shown in the next section.

Results using LMS based algorithm

Generated inputs and outputs are extracted from the data provided in [85]. A network of 10 neurons is selected to identify each nonlinearity. The centers of the RBFNNs are selected using the *k-means* method as before. The spread of the basis function for the four neural networks are optimized to values of 0.15, 0.15, 1.35 and 0.007. The learning rate is optimized to a value of 10^{-8} after hit and trial.

The LMS based algorithm identifies the system very accurately. Mean squared error between the normalized values of original and estimated outputs converges to a final value of 4.2×10^{-2} in around 120 iterations. Once error converges to a final value, the identified model is validated with the rest of the data samples.

Estimates of validated outputs are shown and compared with the original outputs in figures 5.30, 5.31, 5.32 and 5.33. The squared error plot is shown in figure 5.34. Figures 5.35, 5.36, 5.37, and 5.38 show the learnt nonlinearities.

Results using PSO based algorithm

The same data set is used for identification using PSO based algorithm. Again, a network of 10 neurons is selected to identify each nonlinearity. The centers of the RBFNNs are selected using the *k-means* method as before. The spread of the basis function for the four neural networks and the learning rate are optimized to the same values as were used for the LMS case.

PSO based algorithm identifies the system. Once algorithm reaches a final

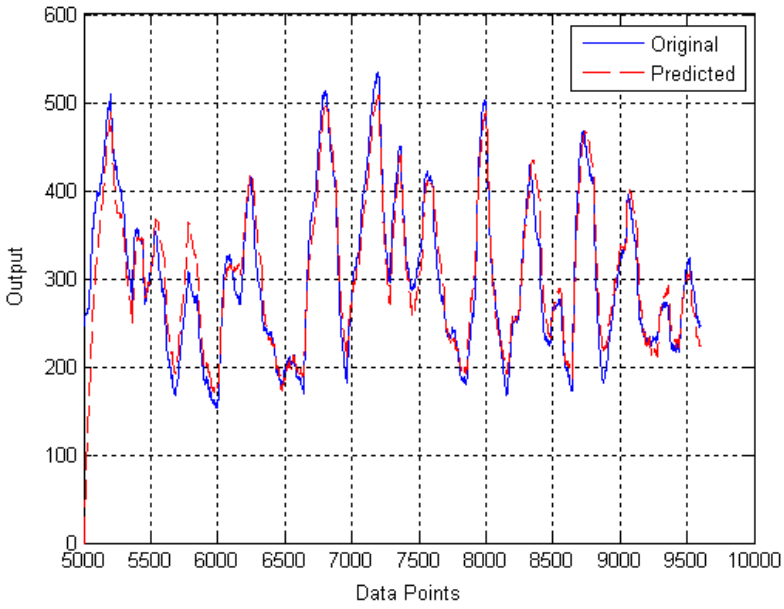

Figure 5.30: Validation result for output y_1 of MIMO example 2 (steam generator) identified by LMS based algorithm

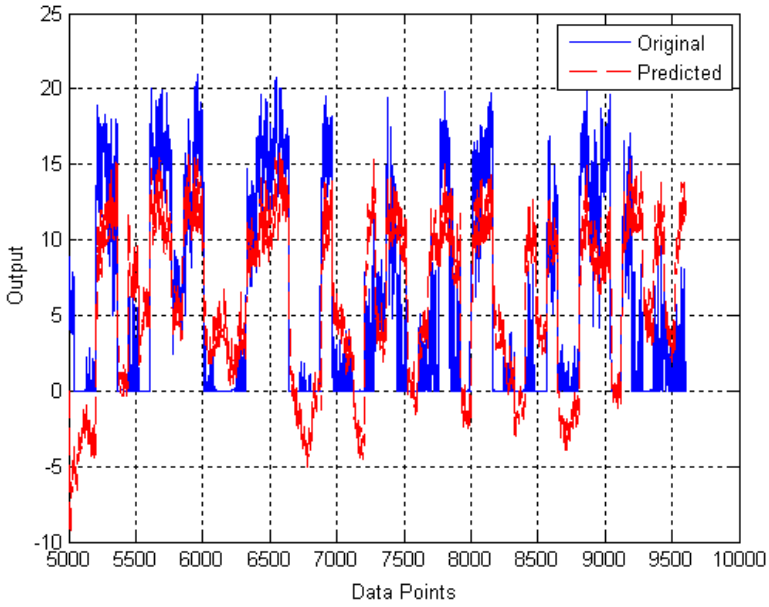

Figure 5.31: Validation result for output y_2 of MIMO example 2 (steam generator) identified by LMS based algorithm

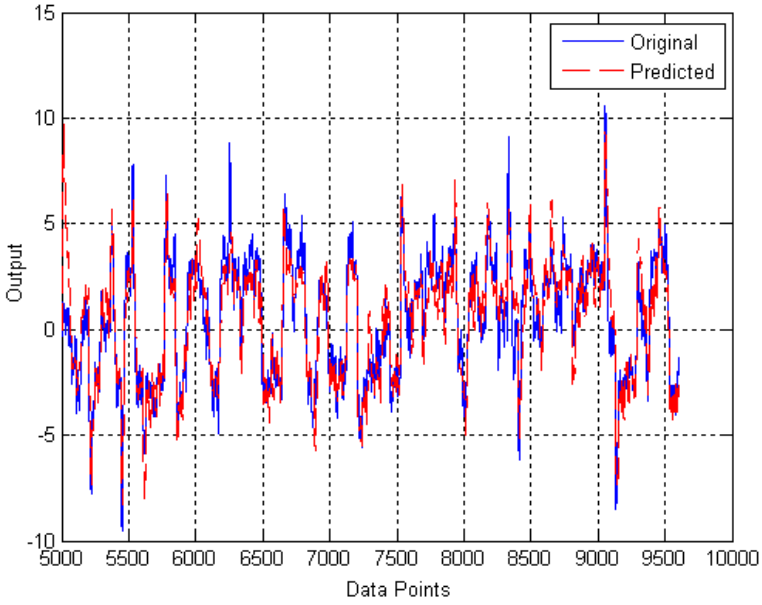

Figure 5.32: Validation result for output y_3 of MIMO example 2 (steam generator) identified by LMS based algorithm

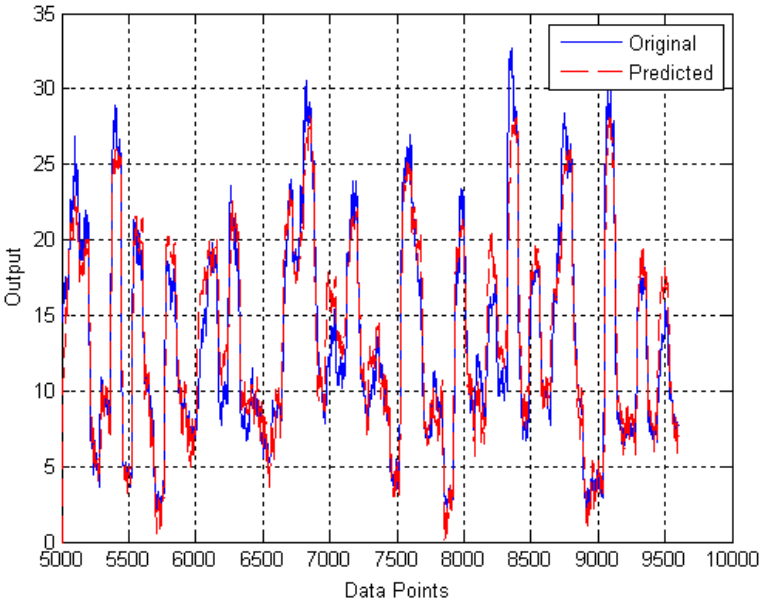

Figure 5.33: Validation result for output y_4 of MIMO example 2 (steam generator) identified by LMS based algorithm

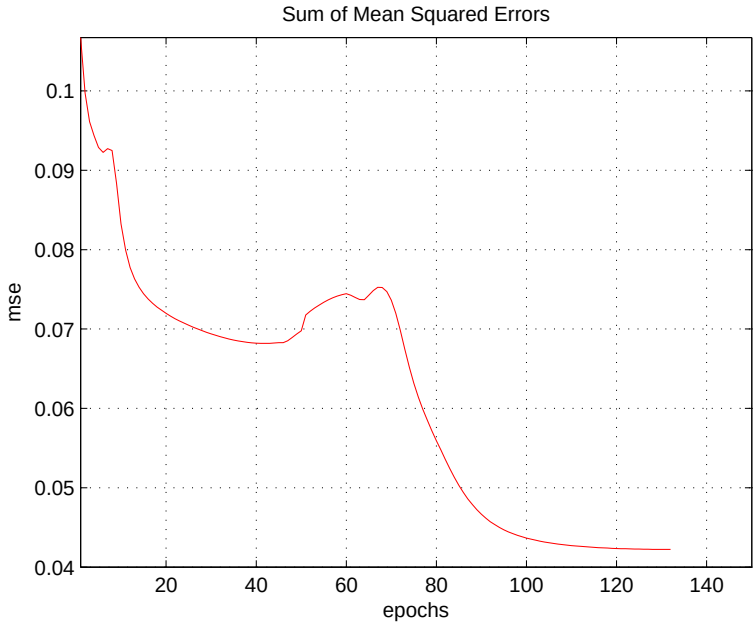

Figure 5.34: Squared error for MIMO example 2 (steam generator) identified by LMS based algorithm

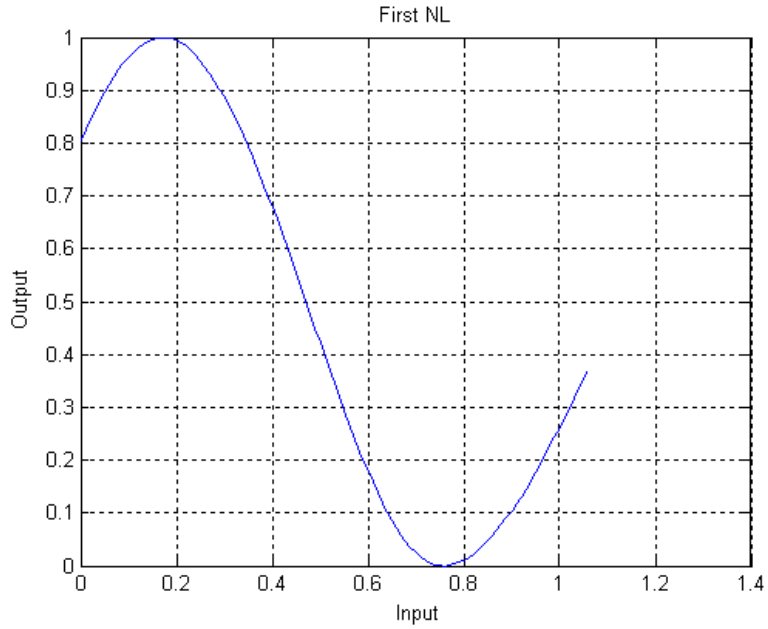

Figure 5.35: First nonlinearity for MIMO example 2 (steam generator) identified by LMS based algorithm

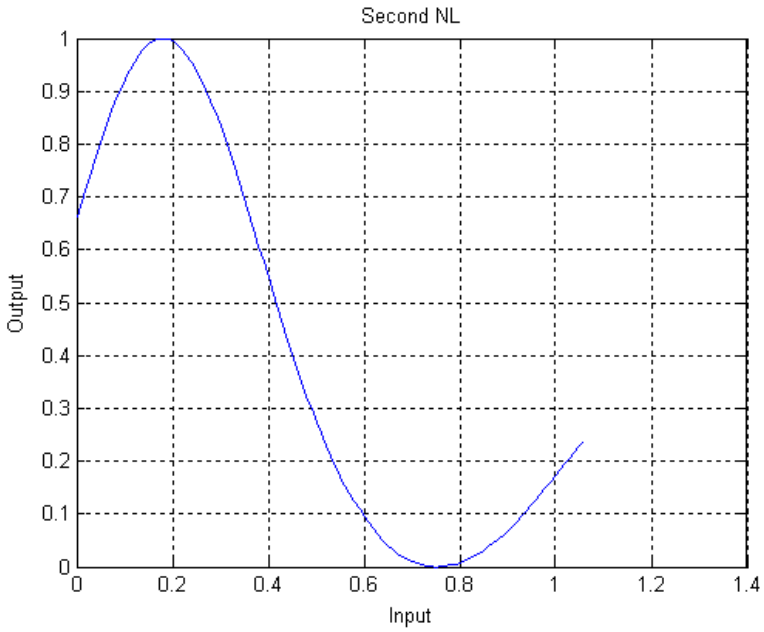

Figure 5.36: Second nonlinearity for MIMO example 2 (steam generator) identified by LMS based algorithm

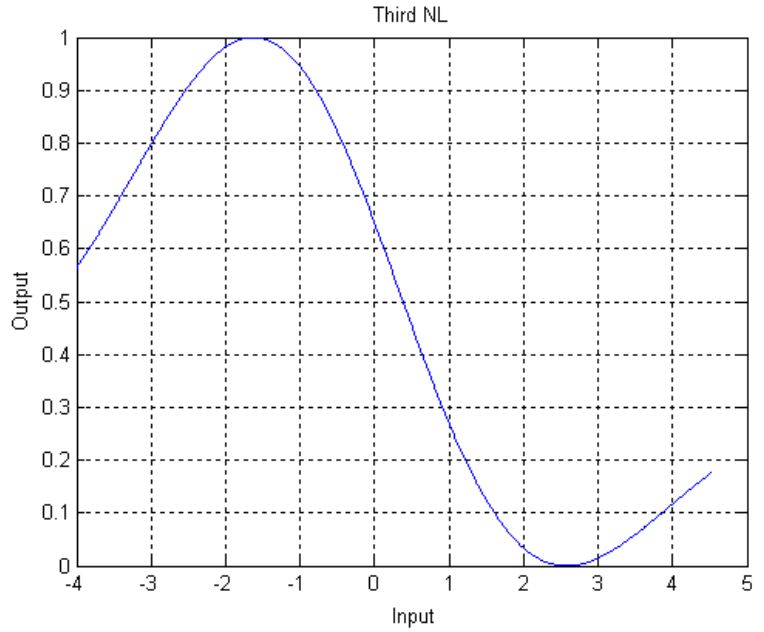

Figure 5.37: Third nonlinearity for MIMO example 2 (steam generator) identified by LMS based algorithm

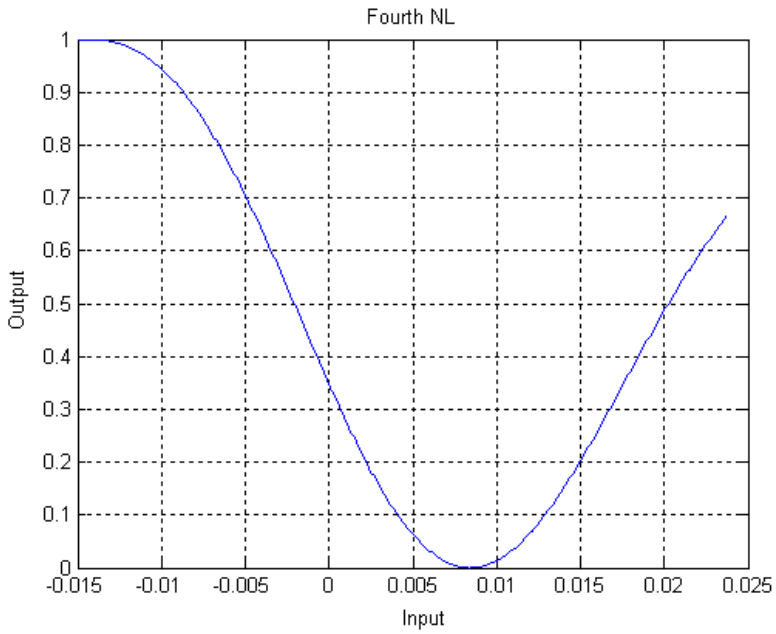

Figure 5.38: Fourth nonlinearity for MIMO example 2 (steam generator) identified by LMS based algorithm

state, the mean squared error converges around a final value of 5×10^{-2} in around 40 iterations. After 40 iterations, the mean square error remains roughly around the same final value, and no further improvement is seen. Once error converges to this final value, the identified model is then validated with the rest of the data samples.

Figures 5.39, 5.40, 5.41, and 5.42 show the validation results. figure 5.43 shows the mean squared error between the normalized values of estimated and original outputs. Figures 5.44, 5.45, 5.46, and 5.47 show the detected nonlinearities in the system, which are almost the same as the nonlinearities detected by LMS based algorithm.

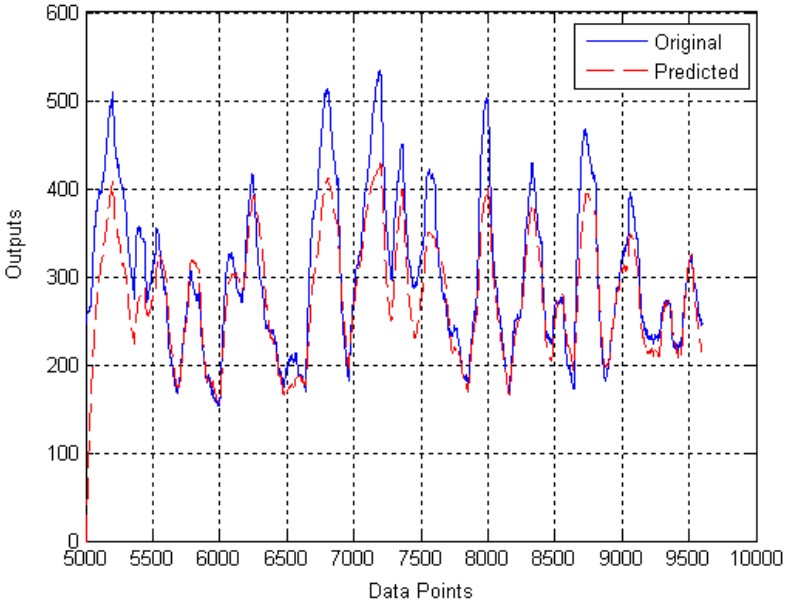

Figure 5.39: Validation result for output y_1 of MIMO example 2 (steam generator) identified by PSO based algorithm

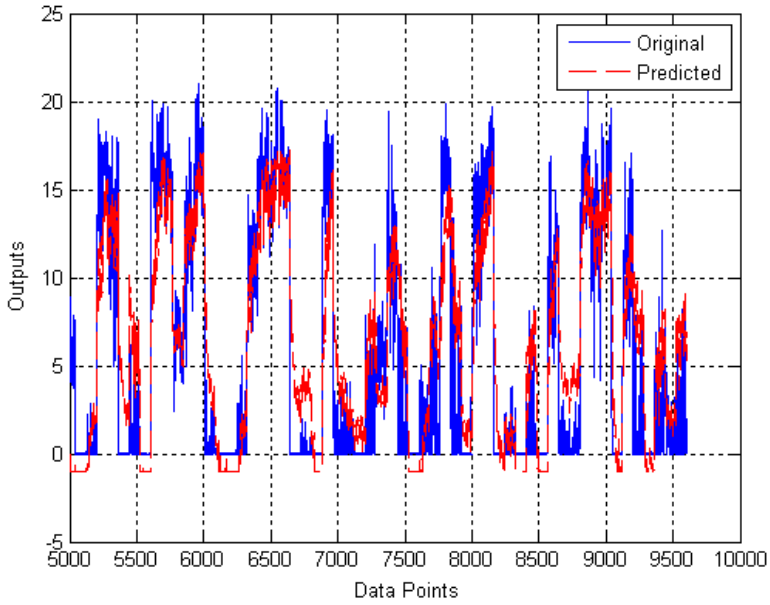

Figure 5.40: Validation result for output y_2 of MIMO example 2 (steam generator) identified by PSO based algorithm

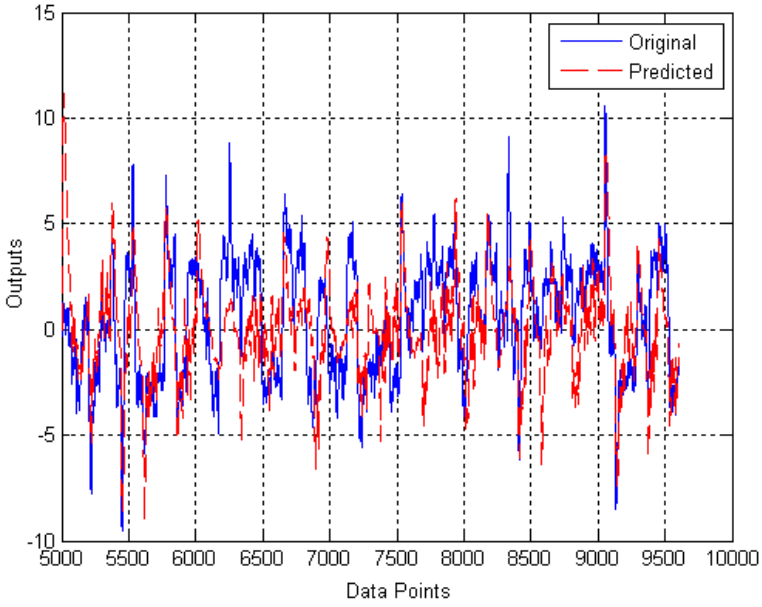

Figure 5.41: Validation result for output y_3 of MIMO example 2 (steam generator) identified by PSO based algorithm

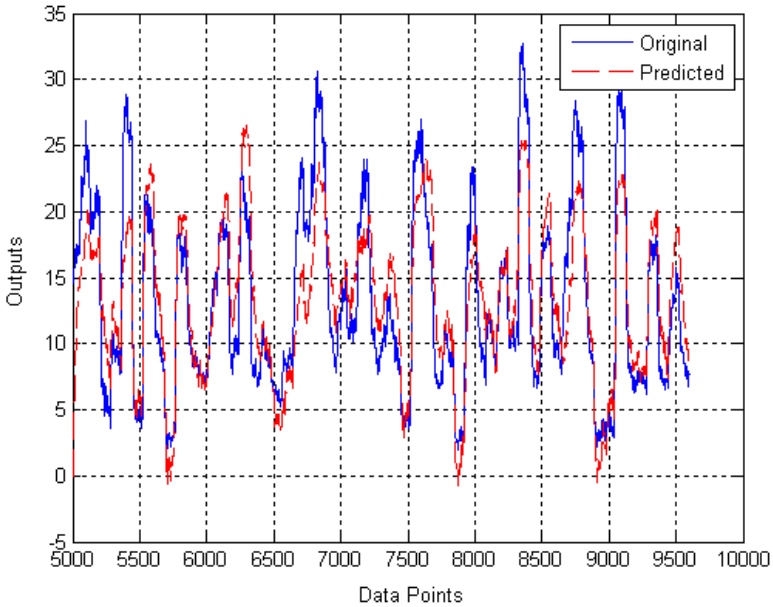

Figure 5.42: Validation result for output y_4 of MIMO example 2 (steam generator) identified by PSO based algorithm

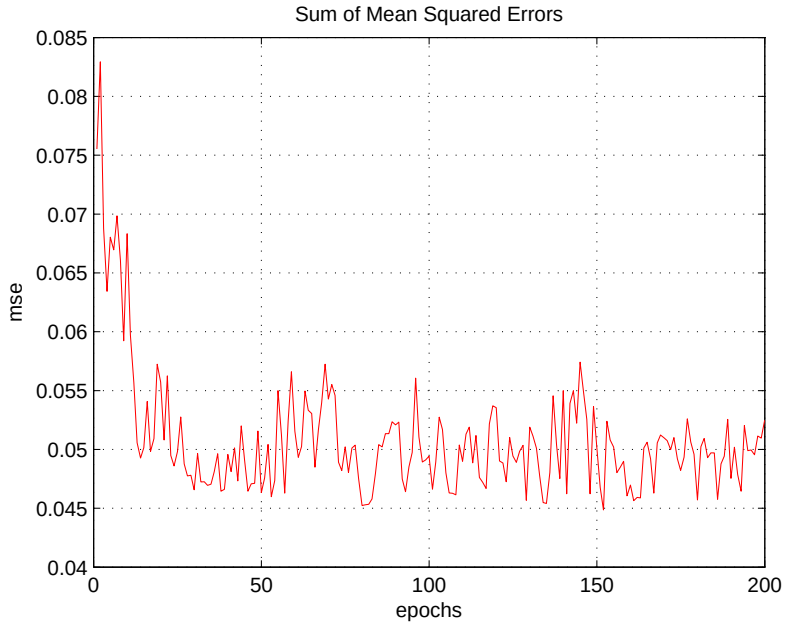

Figure 5.43: Squared error for MIMO example 2 (steam generator) identified by PSO based algorithm

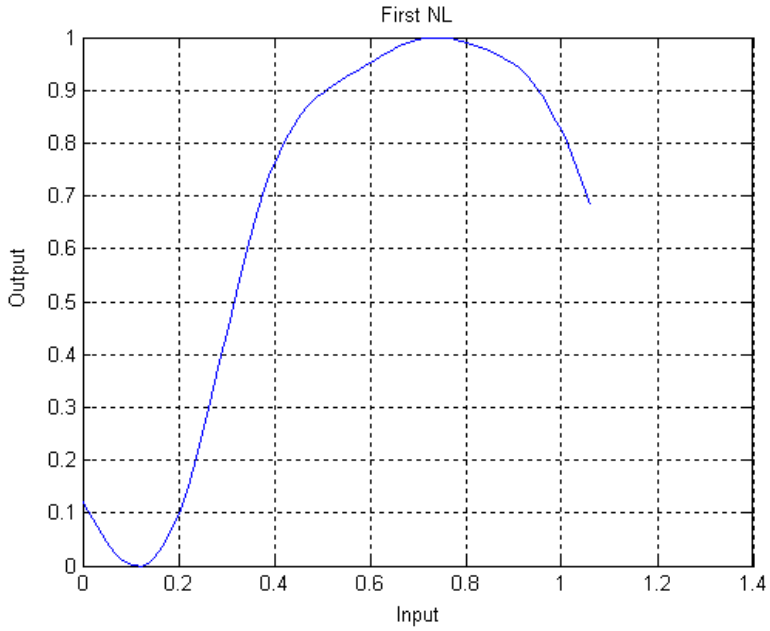

Figure 5.44: First nonlinearity for MIMO example 2 (steam generator) identified by PSO based algorithm

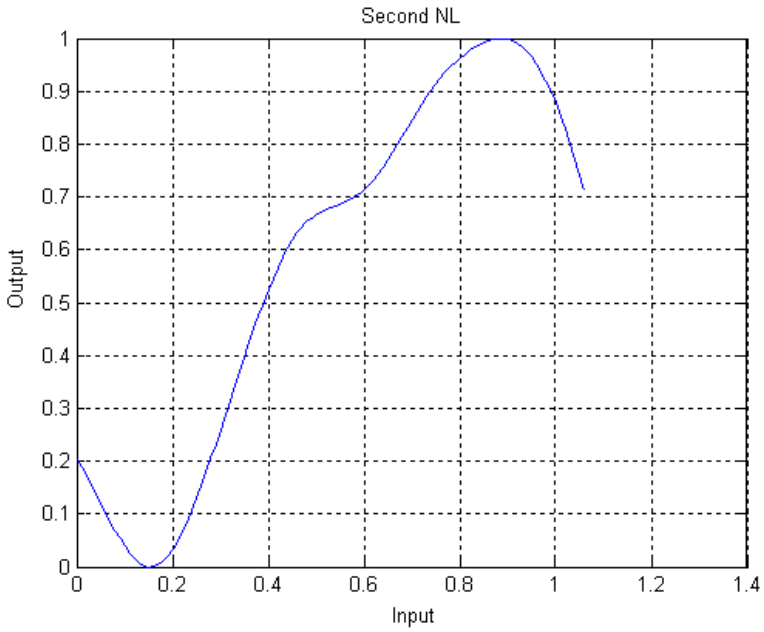

Figure 5.45: Second nonlinearity for MIMO example 2 (steam generator) identified by PSO based algorithm

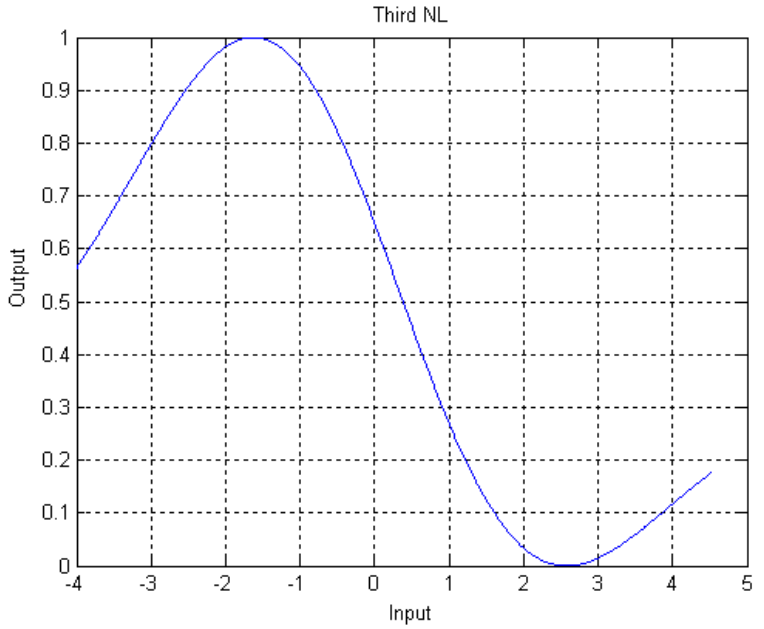

Figure 5.46: Third nonlinearity for MIMO example 2 (steam generator) identified by PSO based algorithm

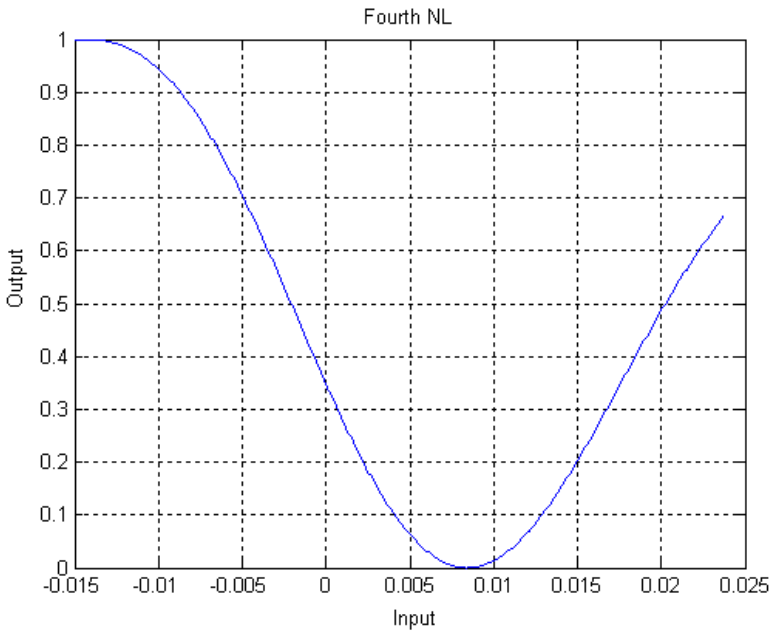

Figure 5.47: Fourth nonlinearity for MIMO example 2 (steam generator) identified by PSO based algorithm

Comparison of results

Both algorithms have identified the steam generator plant. LMS based algorithm has once again showed better results, and has converged to a smaller value of squared error in lesser time. Validation outputs verify this fact too.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusions

This thesis proposes a new method for identification of MIMO hammerstein models. A new identification structure composed of neural network and state-space model has been proposed. Two training algorithms have been developed. The first algorithm uses LMS principle to update neural network weights and subspace identification to update state-space models. The second algorithm uses PSO for estimation of neural network weights and subspace identification for estimation of state space models.

Update equations have been derived for both the SISO and MIMO cases. These training algorithms have then been applied on various examples and promising results have been obtained. All examples have shown consistently good identification

results. The steam generator example has shown that the proposed algorithm is effective on real life industrial plant data as well, which has several disturbances, nonlinearities, high order dynamics, and noise. This shows that the proposed algorithm is immune to noise and can deal with practical real-time examples.

6.2 Recommendation for further work

Based on the knowledge gained during the course of this work, the following recommendations can be made

- Data bracketing can be used for identification of state-space models through subspace identification instead of using all the data points. This will reduce computational complexity considerably.
- Learning rate can be made adaptive. This will surely help convergence towards a better estimate.
- Different enhanced versions of PSO like hybrid-PSO or extended PSO (EPSO) can be used to search for optimum weights of the neural network.

APPENDIX A

PSO

Particle Swarm Optimization is a heuristic search based optimzation technique which exhibits behavior of *swarm intelligence* [78]. Kennedy and Eberhart in [78] described the five basic principles on which swarm intelligence works. These are,

1. The proximity principle: The swarm should be able to carry out simple time and space calculations.

2. The quality principle: The swarm should be able to respond to quality factors in the environment.

3. The principle of diverse response: The swarm should not commit its activities along excessively narrow channels.

4. The principle of stability: The swarm should not change its behavior every time the environment changes.

5. The principle of adaptability: The swarm must be able to change its behavioral mode when its worth the computational price.

PSO differs from other *Evolutionary Algorithms* (EAs) like *Genetic Algorithm* (GA) in small things. The main difference is the way we change the population from one iteration to the next. In GA and other EAs, genetic operators like selection, mutation and crossover are used to change population in one generation to the next, whereas in PSO, the particles are modified according to two formulas. Also, conceptually, in PSO, the particles stay *alive* and the search is *directed*, as every particle position is updated in the direction of the optimal solution where as in EA, the individuals are replaced in each generation. A fundamental conceptual difference is that in GA and other EAs optimal solution is obtained through *competitive search* whereas in PSO, it is reached through *cooperative search*.

Due to these and several other factors, PSO differs from other EAs in terms of performance. PSO is a more robust and fast algorithm that can solve most complex and nonlinear problems. It is very effective in finding global minimum to an optimization problem. It can generate a better solution within lesser time and exhibits stable convergence characteristic than other stochastic methods.

As in other EAs, PSO has initially a population of particles. These particles "evolve" by cooperation and competition among themselves through successive generations. Each particle adjusts its "position" according to its own experience as well as by the experience of neighboring particles.

Each particle is treated as a point in D-dimensional space. The ith particle is represented as

$$X_i = (x_{i1}, x_{i2}, ..., x_{iD}) \tag{A.1}$$

The best previous position (the position giving the best fitness value) of any particle is recorded and represented as

$$P_i = (p_{i1}, p_{i2}, ..., p_{iD}) \tag{A.2}$$

Similarly, the position change (velocity) of each particle is

$$V_i = (v_{i1}, v_{i2}, ..., v_{iD}) \tag{A.3}$$

The particles are manipulated according to the following equations

$$V_i^{n+1} = w * V_i^n + c_1 * r_{i1}^n * (P_i^n - X_i^n) + c_2 * r_{i2}^n * (P_g^n - X_i^n) \tag{A.4}$$

$$X_i^{n+1} = X_i^n + x * V_i^{n+1} \tag{A.5}$$

For multi-modal problems there is a possibility that the algorithm gets stuck in local optimum. This is called *premature convergence*. Avoiding this means looking at all possible local optima before deciding on the global optimum. Several different version of the algorithm like Hybrid-PSO were proposed to deal with this difficulty.

An Extended PSO (EPSO) algorithm was proposed by Xu in 2005. This algorithm uses local as well as global best positions for the calculation of the particles velocity at each iteration [80]. Thus the new equation for velocity becomes:

$$V_i^{n+1} = K.(V_i^n + c_1 * r_{i1}^n * (P_i^n - X_i^n) + c_2 * r_{i2}^n * (P_l^n - X_i^n) + c_3 * r_{i3}^n * (P_g^n - X_i^n)) \tag{A.6}$$

where K is the constriction factor. This algorithm combines the advantages of *global best solutions* and *local best solutions* together.

APPENDIX B

N4SID NUMERICAL ALGORITHM FOR SUBSPACE IDENTIFICATION

Subspace identification constructs state space models from input-output data by estimating first the Kalman filter states from input-output data, and then obtaining the system matrices. As mentioned in section 2.3, a typical state space representation of a system is given by,

$$x(t+1) = Ax(t) + Bu(t) + w(t) \tag{B.1}$$

$$y(t) = Cx(t) + Du(t) + z(t) \tag{B.2}$$

where $u(t) \epsilon \Re^m$ and $y(t) \epsilon \Re^l$ are the vectors for m inputs and l outputs of the system observed at discrete time instant t. Vector $z(t) \epsilon \Re^{l \times 1}$ and $w(t) \epsilon \Re^{n \times 1}$

are zero mean, white noise sequences called measurement noise and process noise respectively.

N4SID stands for *Numerical algorithm for Subspace State Space System Identification*. As proposed by Peter Van Overschee and Bart De Moor in [71], the main steps of N4SID algorithm are:

- Determination of model order n and a kalman filter state sequence estimates $\hat{x}_i \quad \hat{x}_{i+1} \quad \cdots \quad \hat{x}_{i+j}$. These are found by first projecting row spaces of data block Hankel matrices, and then applying a singular value decomposition.

- Solution of a least squares problem to obtain the state space matrices *A, B, C, and D.*

B.1 Mathematical Tools and Notations

B.1.1 Block Hankel Matrices and State Sequences

Block Hankel matrices can be easily constructed from the given input-output data. Matrix $U_{0|i-1}$ is defined as

$$U_{0|i-1} = \begin{bmatrix} u_0 & u_1 & u_2 & \cdots & u_{j-1} \\ u_1 & u_2 & u_3 & \cdots & u_j \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ u_{i-1} & u_i & u_{i+1} & \cdots & u_{i+j-2} \end{bmatrix} \tag{B.3}$$

This is known as Input block Hankel matrix. The number of block rows i is selected as larger than the maximum order i.e. $i \geq n$. The number of columns j is typically equal to *s-2i+1* signifying that all available s data should be used. j should always be larger than *2i-1*.

From here on the following input matrices notations are used,

$$U_p = U_{0|i-1} \tag{B.4}$$

$$U_f = U_{i|2i-1} \tag{B.5}$$

where U_p and U_f refer to matrices of *past* and *future* inputs. Similarly matrices U_p^+ and U_f^- are defined by shifting the border between past and future one block row down. Therefore,

$$U_p^+ = U_{0|i} \tag{B.6}$$

$$U_f^- = U_{i+1|2i-1} \tag{B.7}$$

Similar definitions hold for the block Hankel matrices with the output vectors, denoted by Y_p and Y_f.

Combination of these inputs and outputs are used as regressors and are defined as

$$W_p = W_{0|i-1} = \begin{bmatrix} U_{0|i-1} \\ Y_{0|i-1} \end{bmatrix} \tag{B.8}$$

The state sequence X_i is defined as,

$$X_i = (x_i \quad x_{i+1} \quad \cdots \quad x_{i+j-2}) \quad \epsilon \quad \Re^{n \times j} \tag{B.9}$$

where the subscript i denotes the subscript of the first element of the state sequence.

B.1.2 Extended Observability Matrix

Extended observability matrix Γ_i is used extensively in subspace identification algorithm, and is defined as

$$\Gamma_i = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-1} \end{bmatrix} \quad \epsilon \quad \Re^{li \times n} \tag{B.10}$$

B.1.3 Geometric Tools

If matrices $A\epsilon\Re^{p\times j}$, $B\epsilon\Re^{q\times j}$ and $C\epsilon\Re^{p\times j}$ are given, then for identification $j \geq max(p, q, r)$. This section takes a look at a few geometric tools that are used in linear algebra and which are of relevance to subspace methods also.

Orthogonal Projections

The orthogonal projection of the row space of A onto the row space of B is denoted by A/B and its matrix representation is

$$A/B = A.\Pi_B = AB^T(BB^T)^\dagger B \tag{B.11}$$

where † denotes the Moor-Penrose pseudo-inverse of the matrix and Π_B denotes the operator that projects the row space of a matrix onto the row space of the matrix B. Similarly $A/B^\perp$ denotes projection of the row space of A onto $B^\perp$. Where $B^\perp$ is the orthogonal complement of the row space of B

$$A/B^\perp = A.\Pi_B = A - A/B = A(I_j - \Pi_B) \tag{B.12}$$

Combination of projections Π_B and $\Pi_{B^\perp}$ decomposes a matrix A into two matrices whose row spaces are orthogonal

$$A = A\Pi_B + A\Pi_{B^\perp} \tag{B.13}$$

The matrix representation of these projections are computed by using RQ decomposition of $\begin{bmatrix} B \\ A \end{bmatrix}$. This matrix is the numerical matrix version of the Gram-Schmidt orthogonalization procedure. Let A and B be matrices of full rank and let RQ decomposition of $\begin{bmatrix} B \\ A \end{bmatrix}$ be denoted by

$$\begin{bmatrix} B \\ A \end{bmatrix} = RQ^T = \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \tag{B.14}$$

where $R \epsilon \Re^{(p+q)\times(p+d)}$ is lower triangular, with $R_{11} \epsilon \Re^{q\times q}$, $R_{21} \epsilon \Re^{p\times q}$, $R_{22} \epsilon \Re^{p\times p}$ and $Q \epsilon \Re^{j\times(p+q)}$ are orthogonal i.e.

$$Q^T Q = \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} = \begin{bmatrix} I_q & 0 \\ 0 & I_q \end{bmatrix}$$

Then, the matrix representations of the orthogonal projections can be written as

$$A/B = R_{21} Q_1^T \tag{B.15}$$

$$A/B^{\perp} = R_{22} Q_2^T \tag{B.16}$$

Oblique Projections

A matrix A can also be decomposed as a linear combination of the rows of two nonorthogonal matrices B and C and of the orthogonal complement of B and C.

This can be written as:

$$A = R_B B + R_C C + R_{B^\perp C^\perp} \begin{bmatrix} B \\ C \end{bmatrix}^\perp \tag{B.17}$$

The matrix $R_C C$ is defined as the oblique projection of row space of A along the row space of B into the row space of C

$$A/_B C = R_C C \tag{B.18}$$

Oblique projection means projecting the row space of A orthogonally into the joint row space of B and C and decomposing the result along the row space of B and C

If the RQ decomposition of $\begin{bmatrix} B \\ C \\ A \end{bmatrix}$ is given by Eq. B.19

$$\begin{bmatrix} B \\ C \\ A \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \end{bmatrix}$$

then the matrix representation of the orthogonal projection of the row space of A onto the joint row space of B and C is equal to

$$A/\begin{bmatrix} B \\ C \end{bmatrix} = \begin{bmatrix} R_{31} & R_{31} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \tag{B.19}$$

This can also be written as linear combination of the rows of B and C

$$A/\begin{bmatrix} B \\ C \end{bmatrix} = \begin{bmatrix} R_B B & R_C C \end{bmatrix} = \begin{bmatrix} R_B & R_C \end{bmatrix} \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \tag{B.20}$$

The oblique projection of the row space of A along the row space of B onto the row space of C can thus be computed as

$$A/_B C = R_C C = R_{32} R_{22}^{-1} \begin{bmatrix} R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \tag{B.21}$$

B.2 The Algorithm of N4SID

The N4SID algorithm works in two main steps. First step includes obtaining row space of Kalman filter state sequence from input output data, without any knowledge of the system matrices. Second step involves extraction of system matrices from the state sequence via a least squares problem.

B.2.1 The Kalman State Sequences

State sequence of a combined deterministic−stochastic model can again be obtained from input output data in two steps. First, the future output row space is projected along the future input row space into the joint row space of past input and past output. Singular value decomposition is carried out to obtain the model order, the observability matrix and a state sequence.

Oblique projection

RQ decomposition is used to compute the oblique projection $Y_f/U_f \begin{bmatrix} U_p \\ Y_p \end{bmatrix}$. Let $U_{0|2i-1}$ be the $(2mi \times j)$ and $Y_{0|2i-1}$ the $(2li \times j)$ block Hankel matrices of the input and output observations. Then the RQ decomposition of $\begin{bmatrix} U \\ Y \end{bmatrix}$ is partitioned as follows

$$\begin{bmatrix} U_{0|i-1} \\ U_{i|i} \\ U_{i+1|2i-1} \\ Y_{0|i-1} \\ Y_{i|i} \\ Y_{i+1|2i-1} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 & 0 & 0 & 0 \\ R_{21} & R_{22} & 0 & 0 & 0 & 0 \\ R_{31} & R_{32} & R_{33} & 0 & 0 & 0 \\ R_{41} & R_{42} & R_{43} & R_{44} & 0 & 0 \\ R_{51} & R_{52} & R_{53} & R_{54} & R_{55} & 0 \\ R_{61} & R_{62} & R_{63} & R_{64} & R_{65} & R_{66} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \\ Q_4^T \\ Q_5^T \\ Q_6^T \end{bmatrix} \tag{B.22}$$

The matrix representation of the oblique projection $Y_f/U_f \begin{bmatrix} U_p \\ Y_p \end{bmatrix}$ of the future output row space along the future input row space onto the joint space of past input and past output, is denoted by o_i and is obtained as follows

$$o_i = Y_f/U_f \begin{bmatrix} U_p \\ Y_p \end{bmatrix} = R_{U_p} R_{11} Q_1^T + R_{Y_p} \begin{bmatrix} R_{41} & R_{42} & R_{43} & R_{44} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \\ Q_4^T \end{bmatrix} \quad \text{(B.23)}$$

where

$$\begin{bmatrix} R_{U_p} & R_{U_f} & R_{Y_p} \end{bmatrix} \begin{bmatrix} R_{11} & 0 & 0 & 0 \\ R_{21} & R_{22} & 0 & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ R_{41} & R_{42} & R_{43} & R_{44} \end{bmatrix} = \begin{bmatrix} R_{51} & R_{52} & R_{53} & R_{54} \\ R_{61} & R_{62} & R_{63} & R_{64} \end{bmatrix} \quad \text{(B.24)}$$

from which R_{U_p}, R_{U_f}, and R_{Y_p} can be calculated. The oblique projection $Y_f^-/U_f^- \begin{bmatrix} U_p^+ \\ Y_p^+ \end{bmatrix}$, denoted by o_{i-1} on the other hand, is equal to

$$o_{i-1} = R^+_{U_p} \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} + R^+_{Y_p} \begin{bmatrix} R_{41} & R_{42} & R_{43} & R_{44} & 0 \\ R_{51} & R_{52} & R_{53} & R_{54} & R_{55} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \\ Q_4^T \\ Q_5^T \end{bmatrix}$$

where,

$$\begin{bmatrix} R^+_{U_p} & R^+_{U_p} & R^+_{Y_p} \end{bmatrix} \begin{bmatrix} R_{11} & 0 & 0 & 0 & 0 \\ R_{21} & R_{22} & 0 & 0 & 0 \\ R_{31} & R_{32} & R_{33} & 0 & 0 \\ R_{41} & R_{42} & R_{43} & R_{44} & 0 \\ R_{51} & R_{52} & R_{53} & R_{54} & R_{55} \end{bmatrix} = \begin{bmatrix} R_{61} & R_{62} & R_{63} & R_{64} & R_{65} \end{bmatrix} \tag{B.25}$$

It is assumed that

- The process noise $w(t)$ and measurement noise $v(t)$ are uncorrelated with input $u(t)$
- The input $u(t)$ is persistently exciting of order $2i$
- The number of available data is large, so that j $\rightarrow \infty$

As shown in [71], we know that the oblique projection o_i is equal to the product of the extended observability matrix Γ_i and a sequence of Kalman filter states

$$o_i = \Gamma_i \tilde{X}_i \tag{B.26}$$

Similarly the oblique projection o_{i-1} is equal to

$$o_{i-1} = \Gamma_{i-1} \tilde{X}_{i+1} \tag{B.27}$$

Singular Value Decomposition (SVD)

The singular value decomposition of $R_{U_p} \begin{bmatrix} R_{11} & 0 & 0 & 0 \end{bmatrix} + R_{Y_p} \begin{bmatrix} R_{41} & R_{42} & R_{43} & R_{44} \end{bmatrix}$ is equal to:

$$R_{U_p} \begin{bmatrix} R_{11} & 0 & 0 & 0 \end{bmatrix} + R_{Y_p} \begin{bmatrix} R_{41} & R_{42} & R_{43} & R_{44} \end{bmatrix} \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 S_1 V_1^T \tag{B.28}$$

where $U_1 \epsilon \Re^{li \times n}$, $S_1 \epsilon \Re^{n \times n}$, and $V_1 \epsilon \Re^{li \times n}$. The number of singular values in equation B.28 gives the order of the system. The extended observability matrix Γ_i is chosen as

$$\Gamma_i = U_1 S_1^{1/2} \tag{B.29}$$

and the state sequence $\widetilde{X}_i$ is equal to

$$\widetilde{X}_i = (\Gamma_i)^{\dagger} o_i = S_1^{1/2} V_1^T \begin{bmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \\ Q_4^T \end{bmatrix} \tag{B.30}$$

The *shifted* state sequence $\widetilde{X}_{i+1}$ can be obtained as,

$$\widetilde{X}_{i+1} = (\underline{\Gamma}_i)^{\dagger} o_{i-1} \tag{B.31}$$

where $\underline{\Gamma}_i = \Gamma_{i-l}$ denotes the matrix $\underline{\Gamma}_i$ without the last l rows.

B.2.2 Computing System Matrices

From previous section, the following information has been found

- The order of the system from inspection of the singular values of equation B.28
- The extended observability matrix Γ_i from equation B.29 and the matrix Γ_{i-l} as $\underline{\Gamma}_i$ which denotes the matrix Γ_i without the last l rows
- The state sequences $\tilde{X}_i$ and $\tilde{X}_{i+1}$

The state space matrices A, B, C and D can now be found by solving a set of over determined equations in a least squares sense

$$\begin{bmatrix} \tilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} \begin{bmatrix} \tilde{X} \\ U_{i|i} \end{bmatrix} \tag{B.32}$$

REFERENCES

[1] K. J. *Aström*, and T. Bohlin, "Numerical identification of linear dynamic systems for normal operating records," *Proc. 2nd IFAC Symp. Theory of Self-Adaptive Systems, Teddington*, 1965

[2] B. L. Ho, and R. E. Kalman,"Effective construction of linear state-variable models from input/output functions," *Regelungstechnik*, vol. 14, no. 12, pp. 545 - 548, 1966.

[3] H. Huijberts, H. Nijmeijer, and R Willems, "System identification in communication with chaotic systems," *IEEE Trans. Circuits and Sys*, vol. 47, no. 6, pp. 800 - 808, Jun. 2000.

[4] P. Jazayeri, D. T. Westwick, and W. D. Rosehart, "Identification of nonlinear aggregrate power system loads," *Proc. IFAC Symp.,* vol. 14, pp. 961 - 966, Apr. 2006.

[5] S. A. Billings, and Q. M. Zhu, "Nonlinear model validation using correlation tests," *Int'l. Jnl. Control,* vol. 60, no. 6, pp. 1107 - 1120, Oct. 1994.

[6] U. Forssell, and L. Ljung, "Identification of unstable systems using output error and Box-Jenkins structures," *IEEE Trans. Auto. Control,* vol. 45, no. 1, pp. 137 - 141, Jan. 2000.

[7] F. Previdi, and M. Lovera, "Identification of nonlinear parametrically varying models using separable least squares," *Int'l Jnl. Control,* vol. 77, no. 16, pp. 1382 - 1392, Sep. 2004.

[8] V. Z. Marmarelis, "Identification of nonlinear biological systems using Laguerre expansions of kernels," *Annals. Biomed. Engg.,* vol. 21, pp. 573 - 589, Jun. 1993.

[9] M. M. Mirgagheri, H. Barbeau, and R. E. Kearney, "Instrinsic and reflex contributions to human ankle stiffness: variation with activation level and position," *Exp. Brain Res.,* vol. 135, pp. 423 - 436, Jul. 2000.

[10] T. Soderstorm, and P. Stoica, *System Identification*. Prentice-Hall, 1989.

[11] J. S. Bendat, *Nonlinear Systems Techniques and Applications*. John Wiley and Sons Inc, 1989.

[12] S. A. Billings, "Identification of nonlinear systems - A survey," *IEE Proceedings* vol. 127, no. 6, pp. 272 - 285, Nov. 1980

[13] R. Haber and H. Unbenhauen, "Structure identification of nonlinear dynamic systems - A survey of input-output approaches," *Automatica*, vol. 26, pp. 651 - 677, 1990.

[14] A. B. Gardiner, "Identification of processes containing single-valued nonlinearities," *International Journal of Control* vol. 18, pp. 1029 - 1039, 1973.

[15] R. V. Webb, "Identification of the Volterra kernels of a process containing single-valued nonlinearities," *Electron. Lett.*, vol. 10, pp. 344 - 346, 1974.

[16] D. Brilinger, "The identification of a particular nonlinear time series system," *Biometrica,* vol. 64, pp. 509 - 515, 1977.

[17] S. A. Billings, and S. Y. Fakhouri, "Identification of nonlinear systems using the Wiener model," *Electron. Lett.,* vol. 17, pp. 502 - 504, 1977.

[18] B. Eskinat, and S. H. Johnson, "Use of Hammerstein models in identification of nonlinear systems," *AIChE Journal,* vol. 37, pp. 255 - 268, 1991.

[19] J. C. Gometz and M. Basualdo, "Nonlinear identification of multicomponent batch distillation processes," *IFAC Symposium on Advanced Control of Chemical Processes ADCHEM,* pp. 989 - 994, 2000.

[20] Y. C. Zhu, "Estimation of an N-L-N Hammerstein-Wiener model," *Automatica,* vol. 38, pp. 1607 - 1614, 2002.

[21] A. Karafatis, L. Wang, and W. R. Cluett, "Identification of Wiener-type nonlinear system in a noisy environments," *International Journal of Control,* vol. 66, pp. 923 - 941, 1997.

[22] S. J. Norquay, A. Palazoglu , and J. A. Romagnoli, "Application of Wiener model predictive contro1 (WMPC) to pH neutralization experiment," *IEEE Trans. on Control Systems Technology,* vol. 7, no. 4, pp. 437 - 445, 1999.

[23] R. Haber, and H. Unbehausen, "Structure identification of nonlinear dynamic systems - a survey of input output approaches," *Automatica*, vol. 24, pp. 651477, 1990.

[24] A. Balesinno, A. Landi, M. Ould-Zmirli , and L. Sani, "Automatic nonlinear auto-tuning method for Hammerstein modeling of electrical dnves," *IEEE Trans. on Industrial Electronics*, vol. IES-48, pp. 645 - 655, 2001.

[25] G. Giunta , G. Jacovitti, and A. Nen, "Bandpass nonlinear system identification by higher order cross.correlation," *IEEE Trans. on Signal Processing*, vol. SP - 39, pp. 2092 - 2095, 1991.

[26] R. Emerson, M. Korenberg, and M. Citron, "Identification of complex cell intensive nonlinearities in a cascade mode of cat-visual cortex," *Biological Cybernetics*, vol. 66, pp. 291 - 300, 1992.

[27] K.S. Narendra, and P. Gallman, "An iterative method for the identification of nonlinear systems using hammerstein model," *IEEE Transactions on Automatic Control*, vol. 11, pp. 546 - 550, 1966.

[28] F. Chang, and R. Luus, "A noniterative method for identification using Hammerstein model," *IEEE Transactions on Automatic Control*, vol. 16, pp. 464 - 468, 1971.

[29] A. Billings, and S. Y. Fakhouri, "Identification of non-linear systems using correlation analysis and pseudorandom inputs," *International Journal of Systems Science*, vol. 11, pp. 261 - 279, 1980.

[30] W. Greblicki, and M. Pawlak, "Hammerstein system identification by non-parametric regression estmitation," *International journal of Control*, vol. 45, pp. 343 - 354, 1987.

[31] W. Greblicki, and M. Pawlak, "Identification of Discrete Hammerstein Systems Using Kernel Regression Estimates," *IEEE transactions on Automatic Control*, vol. Ac - 31, no. 1, Jan. 1986.

[32] W. Greblicki, "Non-parametric orthogonal series identification of Hammerstein systems," *International Journal of Systems Science*, vol. 20, pp. 2335 - 2367, 1989.

[33] W. Greblicki, and M. Pawlak, "Dynamic System Identification with Or der Stat istics," *IEEE Transactions on Information Theory*, vol. 40, no. 5, Sep. 1994.

[34] S. Rangan, G. Wolodkin, and K. Poolla, "New results for Hammerstein System identification," *IEEE 34th conference on Decision and Control*, vol. 1, pp. 697 - 702, Dec. 1995.

[35] L. Sun, W. Liu, and A. Sano, "Least Squares Identification Of Hammerstein Model Based On Over-Sampling Scheme," *IEE UKACC International Conference on Control '96*, Conference Publication no. 427, Sep. 1996.

[36] C. Jonathan, and A. M. Zoubir, "Identification of a class of multiple input-output nonlinear systems driven by sationary non-gaussian processes," *Eighth IEEE Signal Processing Workshop on Statistical Signal and Array Processing (SSAP'96)*, pp. 379 - 382, 1996.

[37] Y. Zhu, and T. Ji, "Idetification of hammerstein models for control," *37th IEEE conference on Decision & Control*, 2000.

[38] C. Juan, and E. Baeyens, "Identification of multivariable Hammerstein systems using rational orthonormal bases," *IEEE 39th conference on Decision and Control*, vol. 1, pp. 2849 - 2854, Dec. 2000.

[39] D. T. Westwick, and R. E. Kearney, "Identification of a Hammerstein Model of the Stretch Reflex EMG using Separable Least Squares," *22nd Annual EMBS International Conference 2000, Chicago IL*

[40] T. Hatanaka, and K. Uosaki, "Hammerstein model identification method based on genetic programming," *IEEE Instrumentation and measurement Technology conference*, pp. 1430 - 1435.

[41] J. Voros, "Recursive Identification of Hammerstein Systems With Discontinuous Nonlinearities Containing Dead-Zones," *IEEE Transactions On Automatic Control*, vol. 48, no. 12, Dec. 2003.

[42] H. Al-Duwaish, M. Nazmul Karim, and V. Chandrasekar, "Hammerstein model identification by multilayer feedforward neural networks," *International Journal of Systems Science*, vol. 28, no. 1, pp. 49 - 54, 1997.

[43] H. Al-Duwaish, M. Nazmul Karim, "A New Method for the Identification of Hammerstein Model," *Automatica*, vol. 33, no. 10, pp. 1871-1875, 1997.

[44] H. Al-Duwaish, "A Genetic Approach to the Identification of Linear Dynamical Systems with Static Nonlinearities," *International Journal of Systems Science*, vol. 31, no. 3, pp. 307-314, 2000.

[45] P. A. Marchi, L. S. Coelho, and A. R. Coelho, "Comparative Study of Parametric and Structural Methodologies in Identification of an Experimental Nonlinear Process," *IEEE International Conference on Control Applications Hawai, USA*, Aug. 22 - 27, 1999.

[46] M. Boutayeb, and M. Darouach, "Identification of the Hammerstein model in the presence of bounded disturbances," *IEEE Conference on Industrial Technology*, vol. 1, pp. 590 - 594, Jan. 2000.

[47] H. Al-Duwaish, and S. A. Ali, "Hammerstein Model Identification Using Radial Basis Functions Neural Networks," *International Conference of Artificial Neural Networks (ICANN)*, vol. 1, pp. 951 - 956, Aug. 2001.

[48] S. Hassouna, P. Coirault, and R. Ouvrard, "Continuous nonlinear system identification using series expansion," *American Control Conference, Arlington, VA*, Jun. 25 - 27, 2001.

[49] E. W. Bai, and M. Fu, "A Blind Approach to Hammerstein Model Identification," *40th IEEE confrence on Decision and Control Florida, USA*, Dec. 2001.

[50] A. Akramizadeh, A. Ali, and K. Hamid "Nonlinear Hammerstein Model Identification Using Genetic Algorithm," *IEEE International Conference on Artificial Intelligence Systems*, 2002.

[51] D. Luo, and A. Leonessa, "Identification of MIMO Hammerstein Systems with Nonlinear Feedback," *American Control Confrence, Anchorage, AK*, May 2002.

[52] M. Kozek, N. Jovanovic, "Identification of Hammerstein/Wiener Nonlinear Systemswith Extended Kalman Filters," *American Control Conference Anchorage, AK*, May 8 - 10, 2002.

[53] T. Hatanaka, K. Uosaki, and K. Masazumi, "Block Oriented Nonlinear Model Identification by Evolutionary Computation Approach." *IEEE Conference on Control Applications*, vol. 1, pp. 43 - 48, Jun. 2003.

[54] E. W. Bai, "Decoupling the linear and nonlinear parts in Hammerstein Model Identification," *Automatica*, vol. 40, pp. 671-676, 2004.

[55] A. Janczak, *Identification of Nonlinear Systems Using Neural Networks and Polynomial Models - A Block Oriented Approach.* Springer, 2005.

[56] L. Jia, M. Chiu, and S. Ge, "Neuro-fuzzy System Based Identification Method for Hammerstein Processes," *5th Asian Control Conference*, vol. 1, pp. 104 - 111, Jul. 2004.

[57] L. Jia, M. Chiu, Wang and S. Ge, "Adaptive Neuro-Fuzzy Identification Method for Hammerstein Model," *IEEE Conference on Cybernetics and Intelligent Systems*, vol. 2, pp. 937 - 942, 2004.

[58] J. Vieira and A. Mota, "Parameter Estimation of Non-Linear Systems With Hammerstein Models Using Neuro-Fuzzy and Polynomial Approximation Approaches," *Budapest, Hungary*, Jul. 2004 .

[59] T. Hachino, K. Deguchi, and H. Takata, "Identification of Hammerstein Model Using Radial Basis Function Networks and Genetic Algorithm," *5th Asian Control Conference*, vol. 1, pp. 124 - 129, Jul. 2004.

[60] I. Goethals, K. Pelckmans, J. Suykens, and B. D. Moor, "Subspace Identification of Hammerstein Systems Using Least Squares Support Vector Machines," *IEEE Trans. on Automatic Control*, vol. 50, no. 10, pp. 1509 - 1519, Oct. 2005.

[61] H. F. Chen, and X. L. Hu, "Nonparametric Approach to Identification of Hammerstein and Wiener Systems," *International Conference on Control and Automation, Budapest, Hungary*, vol. 1, pp. 59 - 64, Jun. 2005.

[62] W. Greblicki, "Continuous-Time Hammerstein System Identification From Sampled Data," *IEEE Trans. On Automatic Control*, vol. 51, no. 7, pp. 1195 - 1200, Jul. 2006.

[63] B. I. Ikharia, and D. T. Westwick, "Identification Of Time-Varying Hammerstein Systems Using A Basis Expansion Approach," *Canadian Conference on Electrical and Computer Engineering*, pp. 1858 - 1861, May 2006.

[64] W. Lin, H. Zhang, and P. Liu, "A New Identification Method for Hammerstein Model Based on PSO," *IEEE International Conference on Mechatronics and Automation*, pp. 2184 - 2188, Jun. 2006.

[65] Z. Wenxiao, "Identification for Hammerstein Systems Using Extended Least Squares Algorithm," *26th Chinese Control Conference*, pp. 241 - 245, Jul. 2007.

[66] X. Hong, and R. J. Mitchell, "Hammerstein model identification algorithm using Bezier−Bernstein approximation," *Control Theory & Applications*, vol. 1, no. 4, pp. 1149 - 1159, Jul. 2007.

[67] S. Haykin, *Neural Networks - A Comprehensive Foundation Second Edition.* Prentice-Hall, 1999.

[68] S . Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks," *IEEE Trans. On Neural Networks*, vol. 2, no. 2, Mar. 1991.

[69] V. Overschee, and B. D. Moor, *Subspace Identification for Linear Systems: Theory - Implementation - Applications.* Kluwer Academic Publishers, 1996.

[70] V. Overschee, and B. D. Moor, "A unifying theorem for three subspace system identification algorithms," *Automatica*, vol. 31, no. 12, pp. 1853 - 1864, 1995.

[71] V. Overschee, and B. D. Moor, "N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems," *Automatica*, vol. 30, no. 1, pp. 75 - 93, 1994.

[72] W. E. Larimore, "Canonical variate analysis in identification, filtering and adaptive control," *29th Conference on Decision and Control*, pp. 596 - 604, 1990.

[73] M. Viberg, B. Ottuaten, B. Wahlberg, and L. Ljung, "Performance of Subspace Based State Space Syrtem Identification Methods," *12th WAC World Congress, Sydney, Australia*, vol. 7, pp. 369 - 372, Jul. 1993.

[74] M. Jansson, and B. Wahlberg, "On consistency of subspace methods for system identification," *Automatica*, vol. 34, no. 12, pp. 1507 - 1519, 1998.

[75] M. Verhaegen, P. Dewilde, "Subspace model identification, part 1: The output-error state-space model identification class of algorithms," *International Journal of Control*, vol. 56, pp. 1187 - 1210, 1992.

[76] V. Overschee, and B. D. Moor, "Closed Loop Subspace System Identification," *36th Conference on Decision & Control San Diego, Califomla, USA*, Dec. 1997.

[77] S. Amjad, "Closed Loop Identification For Model Predictive Control: A Case Study," *King Fahd University of Petroleum & Minerals*, Oct. 2003

[78] J. Kennedy, and R. C. Eberhart, *Swarm Intelligence*. Academic Press, 2001.

[79] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *IEEE International Conference on Neural Networks, Perth, Australia*, vol. 4, pp. 1942 - 1948, 1995.

[80] J. Xu and Z. Xin, "An Extended Particle Swarm Optimizer," *19th IEEE International Parallel and Distributed Processing Symposium,* 2005.

[81] B. Widrow et al, "Adaptive noise cancelling: Principle and applications," *IEEE* vol. 63, no. 12, pp. 1692 - 1716, Dec. 1975.

[82] L. Jia, M. S. Chiu, and S. S. Ge, "Neuro-fuzzy System Based Identification Method for Hammerstein Processes," *5th Asian Control Conference*, vol. 1, pp. 104 - 111, Jul. 2004.

[83] G. Pellegrinetti, and J. Benstman, "Nonlinear Control Oriented Boiler Modeling -A Benchamrk Problem for Controller Design," *IEEE Tran. Control Systems Tech.*, vol.4, no.1, Jan 1996

[84] J. Espinosa, and J. Vandewalle, "Predictive Control Using Fuzzy Models Applied to a Steam Generating Unit," *FLINS 98 3rd. International Workshop on Fuzzy Logic Systems and Intelligent Technologies for Nuclear Science and Industry*

[85] B. D. Moor (ed.), "DaISy: Database for the Identification of Systems," *Department of Electrical Engineering, ESAT/SISTA, K.U.Leuven, Belgium*

[86] L. Ljung, *System Identification Theory for User, 2nd ed.,* Printice-Hall, 1999.

Vitae

- Syed Zeeshan Rizvi
- Born in Karachi, Pakistan on July 14, 1983
- Received Bachelor of Engineering (B.E.) from N.E.D University of Engineering and Technology, Karachi, Pakistan in 2006.
- Joined King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia as a Research Assistant in September 2006.
- Completed Master of Science (M.S.) in Electrical Engineering in June 2008.
- Email: *szeeshanrizvi@gmail.com*