# IDENTIFICATION OF HAMMERSTEIN AND WIENER MODELS USING RADIAL BASIS FUNCTIONS NEURAL NETWORKS


BY

SYED SAAD AZHAR ALI


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

December 2001

UMI Number: 1409907

UMI®

UMI Microform 1409907


ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Syed Saad Azhar Ali** under the direction of his thesis advisor 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**

Dr. Hussain N. Al – Duwaish (Chairman)

Dr. Talal O. Halawani (Member)

Dr. Mohammad Mohandes (Member)

Dr. Samir A. Al – Baiyat
(Department Chairman)

Dr. Osama A. Jannadi
(Dean of Graduate Studies)

Date

*Dedicated to*

**My Parents**

**and**

**My Fiancee**

# ACKNOWLEDGEMENTS

*In the name of Allah, the Most Gracious and the Most Merciful*

All praise and glory goes to Almighty Allah (Subhanahu Wa Ta'ala) who gave me the courage and patience to carry out this work. Peace and blessings of Allah be upon His last Prophet Muhammad (Sallulaho-Alaihe-Wassalam) and all his Sahaba (Razi-Allaho-Anhum) who devoted their lives towards the prosperity and spread of Islam.

Acknowledgement is due to King Fahd University of Petroleum and Minerals for providing support for this work.

My deep appreciation and gratitude goes to my thesis advisor Dr. Hussain, N. Al-Duwaish for his constant endeavour, guidance and the numerous moments of attention he devoted throughout the course of this research work. His valuable suggestions made this work interesting and knowledgeable for me.

I extend my deepest gratitude to my thesis committee members Dr. Talal, O. Halawani and Dr. Mohammad Mohandes for their encouragement and cooperation.

Acknowledgement is due to my fellow Wasif Naeem with whom i had informative discussions at various stages of this work.

Special thanks are due to my house mates, colleagues and my friends for their encouragement, motivation and pivotal support. A few of them are Moin, Shafayat Bhai, Ahmer, Raslan, Sajid, Majid, Faisal, Fareed, Ajmal, Asif, Atif, Junaid, Hamid, Shiraz, Kamran and many others, all of whom I will not be able to name here. They made my work and stay at KFUPM very pleasant and joyful.

Special thanks and appreciation go to my uncle Zulfiquar A. Khan who provided a great deal of moral support.

My heartfelt thanks to my days old friends Imran, Yawar, Naheed and Asher. They truly are my great friends, I wish we could be together again.

I would also like to acknowledge the special prayers of Nana Jaan.

I cannot forget the prayers of my late grandmother and my late uncle, Azhar Hussain, who always wished to see me attain a successful place in this life and always believed in me. May their souls rest in peace and may they earn a high place in paradise. (Ameen)

And last but not the least, I would like to thank my parents, brother and sisters, my fiancee and my other family members including all my uncles ,aunts and my loving cousins; from the core of my heart. Their prayers and encouragement always help me take the right steps in life.

# Contents

# List of Tables

# List of Figures

# THESIS ABSTRACT

**Name:** SYED SAAD AZHAR ALI

**Title:** IDENTIFICATION OF HAMMERSTEIN AND WIENER MODELS USING RADIAL BASIS FUNCTIONS

**Degree:** MASTER OF SCIENCE

**Major Field:** ELECTRICAL ENGINEERING

**Date of Degree:** DECEMBER 2001

*A new method is introduced for the identification of particular types of nonlinear systems i.e., Hammerstein and Wiener models. The Hammerstein model consists of a static nonlinearity, followed by a linear dynamic block, while the Wiener model consists of the linear dynamic block followed by the static nonlinearity. The nonlinearity and the linear dynamic part in both the models are identified by using radial basis functions neural network (RBFNN) and autoregressive moving average (ARMA) model, respectively. The new algorithm makes use of the well known mapping ability of RBFNN. Learning algorithms based on least mean squares (LMS) principle are derived for the training of the identification scheme. The proposed algorithms estimate the weights of the RBFNN and the coefficients of ARMA model simultaneously.*

**Keywords**: *Hammerstein, Wiener, RBFNN, ARMA, LMS, SISO, MIMO.*

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS, DHAHRAN.
DECEMBER 2001

# خلاصة الرسالة

| | |
|---|---|
| الإسم: | سيد سعد اظهر علي |
| العنوان: | تعريف نماذج الهمرستين و الوينر باستخدام شبكات الذكاء الصناعى ذات الدوال الاساسية النصف قطرية |
| الدرجة: | الماجستير في العلوم |
| التخصص الرئيسي: | الهندسة الكهربائية |
| تاريخ التخرج: | كانون أول "ديسمبر" ٢٠٠١ |

قدمنا طريقة جديدة للتعرف على أنواع محددة لنظم غير مستقيمة، كنموذج ( Hammerstein) و ( Wiener). نموذج الهمرستين (Hammerstein) يتألف من نظام غير مستقيم ثابت، يتبع بكتلة ديناميكية طولية. بينما نموذج الوينر (Wiener) يتألف من كتلة ديناميكية طولية، تتبع بعدم استقامة ثابتة. عدم الاستقامة الثابتة تعرف باستعمال وظائف أساس نصف قطرية باستخدام الشبكات العصبية (Neural Networks) أو(RBFNN) و الجزء الديناميكي يعرف باستخدام متوسط متحرك ذاتي الارتداد (ARMA). يستفيد الخوارزم الجديد من القاعدة المعروفة جيدًا ذات القدرة على الاقتران (RBFNN). وقد تم اشتقاق خوارزميات تعتمد على مبدأ اقل المتوسطات المربعة (LMS) ذات القدرة على التعلم، لتدريب وظائف التعرف. الخوارزم المقترح يقدر الأوزان في ال (RBFNN) والمُعاملات في محاكات النماذج في ال(ARMA) .

درجة الماجستير في العلوم

جامعة الملك فهد للبترول والمعادن

كانون اول "ديسمبر" ٢٠٠١

# Chapter 1

# Introduction

## 1.1 System Identification

In system identification, the main objective is to find a mathematical relationship between the inputs and outputs, taking into consideration the internal and external disturbance effect. This relationship is called a *model*. Inputs, outputs and if possible, the disturbances in the system are measured and an estimate of the system is found. The identified models can be used in many applications, such as experimental simulations of a systems and controller design of a plant.

To estimate a system, the first step is to build a model structure. A model structure is an idea about the system and the relationship between the input and the output. Real life systems are nonlinear in nature. So in order to identify them, it is suggested to find an operating region where the system acts as linear. However, the linearization does not model the actual system completely for all operating points. And, thus, due to loss of information by ignoring the nonlinearities of the system, the control system designed based on the linear model, maybe unstable or poorly performing.

A significant amount of research has been done on nonlinear system identification and modeling over the years. The study of nonlinear functionals began way back in 1887 by Volterra [1]. The functionals introduced by Volterra were named after him as Volterra kernels and the series was named, Volterra series. Similar to Volterra, Wiener kernels were used by Wiener [2] to identify nonlinear systems. The algorithm adopted by Wiener was used and modified by numerous authors [3]. French and Butz [4] developed a frequency domain method of measuring the Wiener kernels to be used for nonlinear system identification. Identification of nonlinear systems was also performed by the correlation of input signals. Parametric estimation methods of nonlinear systems identification are presented in [5]. In addition to the above mentioned techniques, few others are also discussed in a comprehensive survey done by Billings [6].

In recent years, new techniques have been used to identify the nonlinear systems *e.g.* neural networks [7], genetic algorithms [8], fuzzy logic [9], Extended Kalman filters [10], set membership technique [11], adaptive filters [12], [13], wavelet networks [14], neuro-fuzzy networks [15].

Block-oriented models are one of the major classes of nonlinear systems, which are the main scope of this thesis. The block-oriented approach consists of linear models followed by, or preceded by a static nonlinearity and is classified as Hammerstein and Wiener models depending on the order of the blocks. Block-oriented models provide simple architecture. Moreover, complex structures can be constructed from simple blocks via parallel-series approach [16].

The next section gives an introduction to Hammerstein and Wiener models followed by the literature review.

## 1.2 Hammerstein and Wiener Models

The behavior of many systems can be approximated by a static nonlinearity cascaded with a linear part or vice versa. These models are known as Hammerstein and Wiener models, respectively, as shown in Fig. 1.1 and Fig. 1.2, where, $u(t)$ is the input to the system, $y(t)$ is the output and $x(t)$ is the intermediate nonmeasureable quantity.

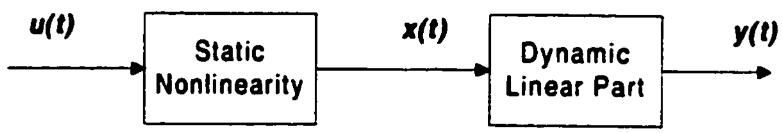


Figure 1.1: Structure of Hammerstein Model.

The Hammerstein and Wiener models are used to model several classes of nonlinear systems. Their flexibility lies in having the nonlinearity entirely separated

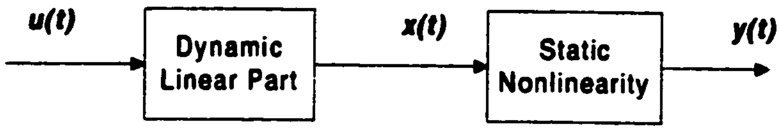


Figure 1.2: Structure of Wiener Model.

from the well known and easily realizable linear parts. This identification gives more insight into the system and simplifies the controller design.

Examples of application of the Hammerstein are, nonlinear filters [17], nonlinear networks [18], detection of signals in non-Gaussian noise [19], nonlinear prediction [20], heat exchangers [21], nonlinear data transmission channels [22], control valves [23], identification of nonlinear systems [24], biological systems [25] like electrically simulated muscle [26], and many others.

The Wiener model has been used in many important applications including pH control [27], fluid control flow [28], identification of biological systems [26], control systems [29] and identification of linear systems with nonlinear sensors [30].

## 1.3 Literature Review

This thesis mainly considers the identification of Hammerstein and Wiener models and proposes a new identification technique. Therefore, the literature review is also classified according to the type of models.

### 1.3.1 Literature on Hammerstein Model Identification

Many identification techniques have been used in the past to identify the linear and nonlinear parts in the Hammerstein model. In 1966, Narendra and Gallman [31], for the first time, identified a nonlinear system by decomposing the nonlinear system into its linear dynamics and static nonlinear blocks. They used an iterative method that gave estimates for the linear and nonlinear parts separately.

Chang and Luus [32] used a non-iterative technique for the identification of nonlinear systems and adopted a batch method.

Billings and Fakhouri [33] identified the nonlinearity using the correlation analysis with pseudo-random inputs. The correlation between the input and output signals was measured and the impulse responses were estimated and updated to minimize the output error.

Greblicki and Pawlak [34]. identified the Hammerstein model by a nonparametric technique, using Kernel regressions estimate.

Hwang and Shyu [35] used finite series expansion method. The system coefficients were expanded in discrete Legendre orthogonal polynomials to identify the system parameters.

Greblicki [36] presented trigonometric as well as Hermite orthogonal series approach to identify the Hammerstein model. Hermite series expansions are found advantageous over the Kernel regressions estimate. Unlike Kernel regressions. the memory requirement in Hermite series is much smaller. The Kernel estimates require as many coefficients as the number of patterns, while the Hermite series expansion requires only a small number of coefficients.

A system comprising of parallel Hammerstein models was identified using polynomial technique by Falkner [37]. This system can accommodate multi-input as well as multiple nonlinearities at the input.

Lang Zi-quang [38] proposed a nonparametric method to identify the Hammerstein model. By using a nonparametric estimation method, a best approximation polynomial of the nonlinear static element was obtained.

Nonparametric deconvolution was used by Greblicki and Pawlak [39] to find the inverse of nonlinearity in frequency domain.

A Kalman filter based recursive algorithm for online identification of the model was put forward by Boutayeb and Darouch [40]. This paper dealt with both the single-input single-output (SISO) and the multi-input single-output (MIMO) Hammerstein systems.

Sundeep et al. [41] concluded that iterative procedure is always preferable to the correlation method. Proposing the impulse response method, the authors guaranteed convergence and concluded that correlation based approaches are essentially unnecessary for the Hammerstein systems.

Jonathan and Zoubir [42] used the Kernels, which replaced the Volterra series, providing a computationally efficient technique. Furthermore, they also used non-

Gaussian input for the identification.

Verhaegen and David [43] used the idea of state space variables. The linear and nonlinear blocks were decomposed into subspaces. They also proposed a subspace model identification for the MIMO Hammerstein systems.

Krzyak and Sasiadek [44] proposed a universal identification method for dynamic nonlinear systems removing the limitations on the class of nonlinearities.

Sequare and Heinz [45] used Hartley modulating functions (HMF). This approach converts the nonlinear differential equations describing the nonlinear system into a Hartley spectrum and finds the estimates in frequency domain.

Zhu [46] revisited the least squares method and modified it for the Hammerstein model. The method was simplified further by model reduction.

Li [47] used Genetic Algorithms (GAs) to identify the Hammerstein model. The GAs are a better approach for a complex system with a large number of parameters. Their solution was based on converting the estimation problem to an optimization problem and minimizing the objective function *e.g.* error at the output. However, there are a number of other parameters that need to be tuned in GAs, like mutation probability, crossover probability, population size, number of generations, etc.

Hammerstein Model was identified using set membership technique by Gustavo and Paolo [11]. The parameters were given a bound within a set, thus they can have any particular value for a value of measurement error, which is also characterized by a set membership context. They suggested to use of any algorithm for the set membership technique. In this paper, the only constraint put on was that the parameters are identifiable if the transfer functions of the linear part are linearly independent.

Gomez and Baeyens [48] identified the Hammerstein systems using rational orthonormal bases. They claimed that their algorithm is numerically robust. The key issue was the representation of the linear part of the system using the orthonormal basis functions which makes it possible allows to write the output equation in linear regressor form. Additionally, the use of rational orthonormal bases allows *a priori* information to be incorporated in the identification process. to improve the estimation accuracy.

Al-Duwaish [21] and Hatanaka *et al.* [49] identified general linear dynamic systems with static non-linearity in particular the Hammerstein model using GAs.

Hammerstein models were identified using MFNN by Duwaish et al. [50]. The MFNN identified the nonlinearity and the linear dynamic part was identified using an ARMA model. A recursive identification algorithm combining recursive least squares(RLS) and back-propagation to identify the Hammerstein model.

### 1.3.2 Literature on Wiener Model Identification

Westwick and Kearney [51] identified the multiple input Wiener system by using a simple cross correlation technique. The estimate of all the linear parts of the system were achieved by using cross correlation technique and the outputs with this estimate were generated for all the systems. The output error was used to update the estimate and then it was iterated until the desired outputs were generated.

Hu and Wang [52] used a three level pseudo-random sequence to estimate the impulse responses of the linear part and the polynomial coefficients of the nonlinearity of the discrete Wiener model.

Duwaish *et al.* [29] presented a neural network approach towards the identification of the Wiener model. The linear part was modelled by an ARMA, where the coefficients were updated using the RLS, and the nonlinearity was estimated by MFNN. The MFNN is updated using the back-propagation algorithm.

A recursive subspace identification algorithm was proposed by Lovera and Verhaegen [53]. The method was found as a by product of the MOESP (multivariable output error state-space model identification) class algorithm.

Chou and Verhaegen [54] used the pre-filtering of both input and output data. This technique is reported to have good results for LTI systems. The authors showed that correlation method can be used for the nonlinear systems to apply data pre-filtering.

Similar to Hammerstein systems, the Wiener model was also identified using the maximum likelihood with a linear regression initialization by Hagenblad and Ljung [55]. The idea is to first parameterize the model. and then parameters are transformed in the form of original system. The inverse of the nonlinear system is parameterized with linear B-splines. Now the estimate of the intermediate signal from the parameters and the B-splines are equated and solved for the parameters.

Rodriguez and Fleming [56] used a multi-objective approach in the genetic programming to identify the nonlinearity in the Wiener model. This method gives a tradeoff between the complexity and the performance of the system.

Greblicki [57] used orthogonal series expansion for the identification. The author also used Kernels regressions in [58] to identify the Wiener model. Greblicki also identified continuous time Hammerstein system [59] and continuous time Wiener system [60].

Hagenblad used the prediction error and expectation maximization method to identify the Wiener model in [23].

Chou and Verhaegen [61] adopted a three step algorithm to identify the Wiener model with process noise. The algorithm was based on cross-correlation and subspace identification. The subspace matrices are estimated first which are used in the estimation of the intermediate signal and together with the output the inverse of the static nonlinearity is estimated. And finally the consistent estimates of the subspace variables are approached.

Yong and Chow [14] presented a hybrid model of wavelets and neural networks as an identification scheme for the Wiener model. The orthogonal scaling and the mother wavelets were combined to obtain the activation functions. The neural network is used to avoid the need of any *apriori* information. The wavelets capture the modes of the nonlinearity by passing the nonlinearity through a time-frequency plane window. The liner part was identified by an ARMA.

A comprehensive analysis of stochastic gradient identification of Wiener systems was done by Celka *et al.* [62].

Anders and Lars [63] identified the time varying Wiener Hammerstein system. They made the use of the extended Kalman filter. Furthermore, to ensure stability, the paper reformulated the algorithm in terms of a nonlinear minimization problem with a quadratic inequality constraint.

Lacy *et al.* [64] identified the Wiener model by minimizing a cost function for standard least squares depending upon the vector of unknown system parameters and the intermediate signal.

## 1.4 Motivation for Present Work

The algorithm presented in [50] and [29] used MFNN to identify the static nonlinearities in the Hammerstein and Wiener models. The back-propagation algorithm is used to update the weights of the MFNN. It is well known in the literature that the convergence of the back-propagation algorithm is very slow compared to LMS when used for training radial basis functions neural networks (RBFNN) [65], [66], [67], [68]. Moreover, the RBFNN has the same universal approximation capabilities as the MFNN [65]. This motivated the use of RBFNN instead of MFNN to model the static nonlinearity. The linear part is modelled by an ARMA model.

## 1.5 Thesis Contributions

This thesis presents a new identification algorithm for the Hammerstein and Wiener models. This work can be stated as an extension of the contributions presented in [50] and [29]. RBFNN are successfully implemented instead of MFNN to identify the static nonlinearities in the Hammerstein and Wiener models. RBFNN appeared as a better choice than MFNN due to faster convergence. The network is also simpler as only one layer is involved for the computations. Simulation shows that lesser number of neurons are required with RBFNN.

The new identification structure for the Hammerstein model consists of an RBFNN followed by an ARMA model. For the Wiener model, the structure consists of an ARMA model followed by an RBFNN. The contributions an be enumerated as follows:

- Training algorithms are derived based on LMS principles to obtain the update equations for SISO Hammerstein and SISO Wiener systems.
- Training algorithms are derived for the MIMO Hammerstein and MIMO Wiener systems with separate nonlinearities *i.e.* each nonlinearity is independent of the other nonlinearities in the system.
- Training algorithms are derived for the MIMO Hammerstein and MIMO Wiener systems with combined nonlinearities *i.e.* there is only one MIMO nonlinear

block in the system and the inputs and outputs of the nonlinear part are all coupled with each other.

- The proposed algorithms identify the Hammerstein and Wiener models in single step, *i.e.* the linear and nonlinear parts are identified simultaneously.

To validate the derived equations, simulations are done for several SISO and MIMO examples. Simulations are also performed for noisy environments and promising results are obtained.

## 1.6 Thesis Organization

This thesis is organized as follows:

Chapter 2, presents the introduction and mathematical representation to the Hammerstein and Wiener models. This chapter also includes the proposed identification structure. The main components of the identification structure *i.e.* the RBFNN and the ARMA model are described.

The contributions begin from Chapter 3, where the proposed training algorithm is developed. In this chapter, the derivation for SISO cases for both the Hammerstein and Wiener models are presented.

Follows Chapter 4, where the proposed algorithm is validated by simulations. Examples for Hammerstein and Wiener models are considered and simulation results with discussions are included in the chapter.

Chapter 5 comprises the derivations for the MIMO cases of Hammerstein and Wiener models. Training algorithms for a general $M$-input $N$-output model are developed.

Chapter 6 includes the example study for MIMO cases and simulation results for each case is briefed.

Chapter 7 presents the conclusions and avenues for future work.

# Chapter 2

# Proposed Identification Structure

This chapter presents the proposed identification structure. Before describing the structure and its components, a brief introduction to the Hammerstein and Wiener models is given, with their mathematical representation. Introduction to the components of the identification structure *i.e.* RBFNN and ARMA is presented towards the end of the chapter.

## 2.1 Hammerstein Model

Hammerstein Model comprises of nonlinear block in cascade with a linear block as shown in Fig. 1.1. The Hammerstein model is used to model several classes of nonlinear systems. Its flexibility lies in having the nonlinearity entirely separated from the common and easily realizable linear parts. The static nonlinear element scales the input $u(t)$ and transforms it to $x(t)$, and the dynamics are modeled by a linear transfer function, whose output is $y(t)$. The Hammerstein model models the nonlinear effects as an input-dependent nonlinear gain. The slope of the nonlinearity at a certain operating point is the instantaneous gain of the system. The Hammerstein model can be described by the following set equations [24].

$$y(t) = \sum_{i=1}^{m} a_i y(t-i) + \sum_{j=0}^{n} b_j x(t-j). \quad (2.1)$$

$$x(t) = f(u(t), \rho). \quad (2.2)$$

Where $u(t)$ is the input to the system and $y(t)$ is the output of the system. $x(t)$ is the nonlinear function of the input and the parameters $\rho$ ($\rho$ corresponds to the weights for the RBFNN). The integers $m$ and $n$ are the order of poles and zeros of the linear system. The quantity, $x(t)$ cannot be measured, but it can be eliminated from the equations. Thus, Eq. 2.1 can be written in the following form in which the intermediate variable $x(t)$ has been removed.

$$\begin{aligned} y(t) &= \frac{B(q^{-1})}{A(q^{-1})} x(t), \\ y(t) &= \frac{B(q^{-1})}{A(q^{-1})} f(u(t), \rho), \end{aligned} \quad (2.3)$$

where $q^{-1}$ is the delay operator and the polynomial $A(q^{-1})$ and $B(q^{-1})$ are.

$$A(q^{-1}) = 1 + a_1 q^{-1} + \cdots + a_n q^{-n}, \quad (2.4)$$

$$B(q^{-1}) = b_1 q^{-1} + \cdots + b_m q^{-m}. \quad (2.5)$$

Now, the Hammerstein system can be modelled entirely in terms of the inputs and the outputs of the system.

## 2.2 Wiener Model

The Wiener model of a nonlinear system is constructed by a nonlinear gain cascaded after a linear subsystem. Similar to the Hammerstein model, Wiener model is also used to model many nonlinear systems. The structure of Wiener model of a nonlinear system has been shown in Fig. 1.2. Mathematically it can be represented by the following equations,

$$y(t) = f(x(t), \rho)). \quad (2.6)$$

$$x(t) = \sum_{i=1}^{m} a_i x(t-i) + \sum_{j=0}^{n} b_j u(t-j), \quad (2.7)$$

$$\text{or} \quad x(t) = \frac{B(q^{-1})}{A(q^{-1})} u(t). \quad (2.8)$$

Where $u(t)$ is the input to the system, $x(t)$ is the intermediate nonmeasureable variable. $y(t)$ is the output and $A(q^{-1})$, $B(q^{-1})$ have been defined in Eq. 2.4 and Eq. 2.5.

The intermediate variable $x(t)$ can be eliminated from observed output $y(t)$ of the system as,

$$y(t) = f\left(\frac{B(q^{-1})}{A(q^{-1})}u(t), \rho\right). \tag{2.9}$$

Therefore, the system is now completely modelled in terms of the input and the output of the system.

## 2.3 Problem Statement

Thus, the problem of the Hammerstein and Wiener model identification is to estimate the coefficients of the linear part $a_1, \ldots, a_n$. $b_o, \ldots, b_m$ and the weights $\rho$ for the nonlinearity. Training algorithm for single input single output (SISO) systems will be developed first, and then will be generalized for multi-input multi-output (MIMO) systems.

A recursive algorithm is to be developed that updates the weights of the RBFNN and the coefficients of the ARMA with each pair of the input-output data, such that the output error is minimized. This makes it an optimization problem to achieve a predefined performance index. Therefore, to minimize the error a gradient descent algorithm can be selected. Literature depicts LMS algorithm to be a simple and easy to implement algorithm for the above said problem.

## 2.4 Proposed Identification Structure

The proposed identification algorithm is also block-oriented and comprises of corresponding linear and nonlinear blocks. The proposed identification structure makes use of the universal approximating capabilities of RBFNN for the static nonlinear block and an ARMA for the linear block.

Identification structure for the Hammerstein model consists of an RBFNN in series with an ARMA model, as shown in Fig. 2.1, and an ARMA model in series with RBFNN for Wiener systems as shown in Fig. 2.2.

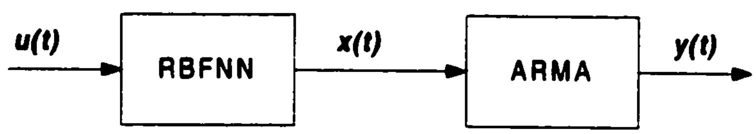


Figure 2.1: RBFNN/ARMA identification structure for Hammerstein model.

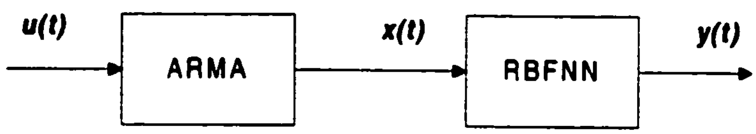


Figure 2.2: ARMA/RBFNN identification structure for Wiener model.

### 2.4.1 Radial Basis Function Neural Networks

RBFNN is a type of feed forward network. They are used in a wide variety of contexts such as function approximation. pattern recognition and time series prediction.

Networks of this type have the universal approximation property. In these networks the learning involves only one layer with lesser computations. This results in reduction in the training time in comparison with MFNN, that uses back propagation algorithm to update the weights of all the layers. These features make RBFNN attractive in many practical problems.

A SISO RBFNN is shown in Fig. 2.3. It consists of an input node $u(t)$, a hidden layer with $n_o$ neurons and an output node $x(t)$.

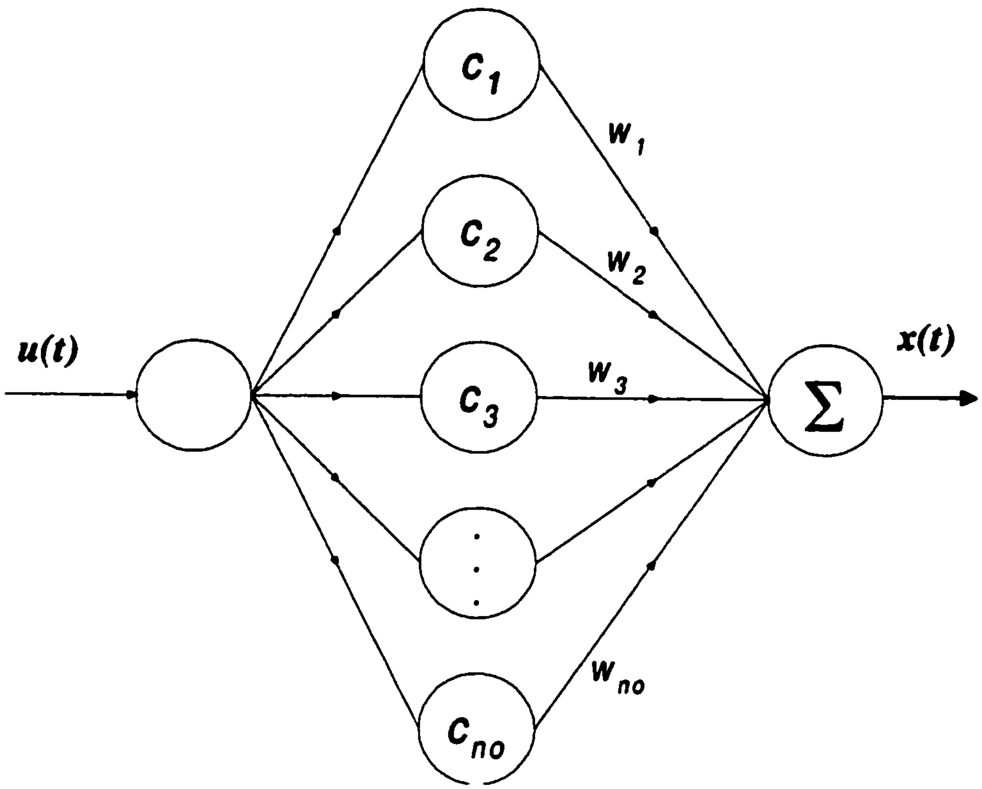


Figure 2.3: A general RBFNN network

Each of the input node is connected to all the nodes in the hidden layer through unity weights (direct connection). While each of the hidden layer nodes is connected to the output node through some weights $w_1, \ldots, w_{n_o}$. Each neuron finds the distance, normally applying Euclidean norm, between the input and its centre and passes the resulting scalar through a non-linearity. So the output of the hidden neuron is given by $\phi(\|u(t) - c_i\|)$, where $n_o$ is the number of hidden layer nodes (neuron), $u(t)$ is the input, $c_i$ is the centre of $i^{th}$ hidden layer node where $i = 1, 2, \ldots, n_o$, and $\phi(\cdot)$ is the nonlinear basis function. Normally this function is taken as a Gaussian function of width $\beta$. The output ($x(t)$) is a weighted sum of the outputs of the hidden layer, given by

$$
\begin{aligned}
x(t) &= W\Phi(t), \\
x(t) &= \sum_{i=1}^{n_o} w_i \phi(\|u(t) - c_i\|).
\end{aligned}
$$

$$
\begin{aligned}
\text{where} \quad W &= [w_1 \; w_2 \; \ldots \; w_{n_o}], \\
\text{and} \quad \Phi(t) &= [\|u(t) - c_1\| \; \|u(t) - c_2\| \; \ldots \; \|u(t) - c_{n_o}\|]^T.
\end{aligned}
$$

$x(t)$ is the output, $w_i$ is the weight corresponding to the $i^{th}$ hidden neuron.

### 2.4.2 ARMA Model

The linear part of the Hammerstein/Wiener model is modeled by an ARMA model, whose output is given by

$$y(t) = \sum_{i=1}^{n} a_i y(t-i) + \sum_{j=0}^{m} b_j u(t-j) \tag{2.10}$$

or in terms of $q^{-1}$ operator

$$y(t) = \frac{B(q^{-1})}{A(q^{-1})} q^{-d} u(t) \tag{2.11}$$

# Chapter 3

# Development of Training Algorithms for SISO Systems

In this chapter the training algorithms for the SISO systems of the Hammerstein and Wiener models are considered. Considering Fig. 2.1 and Fig. 2.2, the objective is to develop a recursive algorithm by which the weights of the RBFNN and coefficients of the ARMA model can be adjusted, such that the set of inputs produces the desired set of outputs. This problem is solved by developing a new parameter estimation algorithm which will be based on the well known LMS principles.

## 3.1 Training Algorithm for Hammerstein Model

The training algorithm is developed based on LMS principle. The proposed identification structure runs in parallel to the actual system as shown in Fig.3.1.

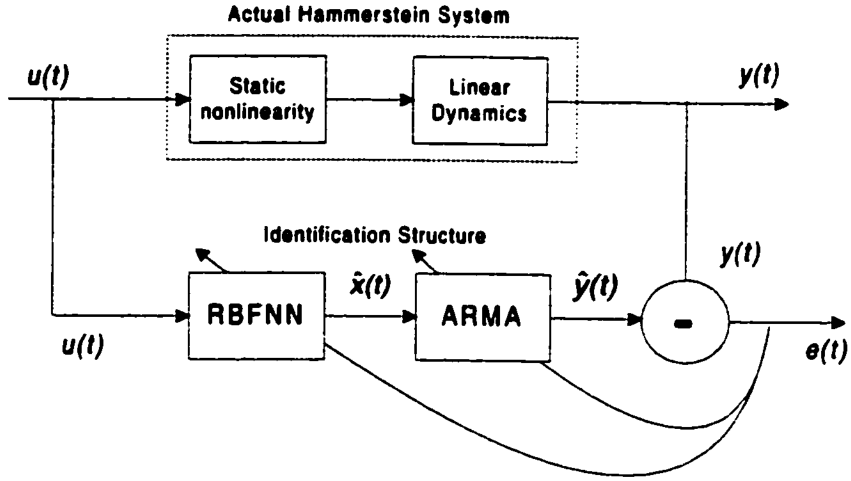


Figure 3.1: Identification of Hammerstein system.

The parameters (weights of RBFNN and the coefficients of ARMA) are updated by minimizing the performance index $I$ given by,

$$
\begin{aligned}
I &= \frac{1}{2}e^2(t), \\
e(t) &= y(t) - \hat{y}(t).
\end{aligned} \tag{3.1}
$$

where $y(t)$ is the actual output of the system and $\hat{y}(t)$ is the estimated output of the Hammerstein Model. The coefficients of the ARMA model and the weights of the RBFNN should be updated in the negative direction of the gradient as.

$$\theta(K+1) = \theta(K) - \alpha \frac{\partial I}{\partial \theta(K)}, \tag{3.2}$$

and

$$W(K+1) = W(K) - \alpha \frac{\partial I}{\partial W(K)}, \tag{3.3}$$

where $\theta = [a_1 \ \ldots \ a_n \ b_o \ \ldots \ b_m]$ is the parameter vector , $W = [w_1 \ w_2 \ \ldots \ w_{n_o}]$ is the weight vector for RBFNN and $\alpha$ is the learning parameter. The variable $K$ is used to show the iteration number of training. This must not be confused with the time variable $t$, as the parameters and the weights are independent of time, once the training process is completed.

Keeping the regressions of the variables in the system in a regression vector $\psi$ as $\psi(t) = [\hat{y}(t-1) \ \ldots \ \hat{y}(t-n) \ \hat{x}(t-d) \ \ldots \ \hat{x}(t-m-d)]$ and finding partial derivatives.

$$\begin{aligned}
\frac{\partial I}{\partial \theta} &= \frac{1}{2}\frac{\partial e^2(t)}{\partial \theta}, \\
&= e(t)\frac{\partial}{\partial \theta}\left(y(t) - \hat{y}(t)\right), \\
&= e(t)\frac{\partial}{\partial \theta}\left(y(t) - \frac{B(q^{-1})}{A(q^{-1})}q^{-d}\hat{x}(t)\right), \\
&= e(t)\frac{\partial}{\partial \theta}\left(y(t) - \left(a_1 q^{-1} + \cdots + a_n q^{-n}\right)\hat{y}(t)\right. \\
&\qquad \left. - \left(b_o + b_1 q^{-1} + \cdots + b_m q^{-m}\right) q^{-d}\hat{x}(t)\right), \\
&= -e(t)\frac{\partial}{\partial \theta}\left(a_1\hat{y}(t-1) + a_2\hat{y}(t-2) + \cdots + a_n\hat{y}(t-n) + \right. \\
&\qquad \left. b_o\hat{x}(t-d) + b_1\hat{x}(t-1-d) + \cdots + b_m\hat{x}(t-m-d)\right).
\end{aligned}$$

$$
\begin{aligned}
&= -e(t)\left[\hat{y}(t-1)\ \hat{y}(t-2)\ \dots\ \hat{y}(t-n)\right.\\
&\qquad\qquad \left.\hat{x}(t-d)\ \hat{x}(t-1-d)\ \dots\ \hat{x}(t-m-d)\right],\\
\frac{\partial I}{\partial \theta} &= -e(t)\psi(t). \qquad (3.4)
\end{aligned}
$$

Now, the gradient in Eq. 3.4 is used to find the updated parameters at $(K+1)^{th}$ instant along with Eq. 3.2. The final parameter update equation will be,

$$\boxed{\theta(K+1) = \theta(K) + \alpha\, e(t)\psi(t).} \qquad (3.5)$$

The partial derivatives for the weights are derived as follows,

$$
\begin{aligned}
\frac{\partial I}{\partial W} &= \frac{1}{2}\frac{\partial e^2(t)}{\partial W},\\
&= e(t)\frac{\partial}{\partial W}\left(y(t) - \hat{y}(t)\right),\\
&= e(t)\frac{\partial}{\partial W}\left(y(t) - \frac{B(q^{-1})}{A(q^{-1})}q^{-d}\hat{x}(t)\right),\\
&= e(t)\frac{\partial}{\partial W}\left(y(t) - \left(a_1q^{-1} + \dots + a_nq^{-n}\right)\hat{y}(t)\right.\\
&\qquad \left. - \left(b_o + b_1q^{-1} + \dots + b_mq^{-m}\right)q^{-d}W\Phi(t)\right),\\
&= -e(t)\left(b_o + b_1q^{-1} + \dots + b_mq^{-m}\right)q^{-d}\Phi(t),\\
\frac{\partial I}{\partial W} &= -e(t)B(q^{-1})q^{-d}\Phi(t).
\end{aligned}
$$

This gradient is used to find the weight update equation, along with Eq. 3.3,

$$\boxed{W(K+1) = W(K) + \alpha\, e(t)B(q^{-1})q^{-d}\Phi(t).} \qquad (3.6)$$

## 3.2 Training Algorithm for Wiener Model

Considering Fig. 2.2, the objective is to develop a recursive algorithm that adjusts the parameters of the ARMA model and the weights of the RBFNN in such a way, that the set of inputs produces the desired set of outputs. This goal is achieved by developing a new parameter estimation algorithm based on the LMS approach. The proposed identification structure runs in parallel with the actual Wiener system as shown in Fig. 3.2.

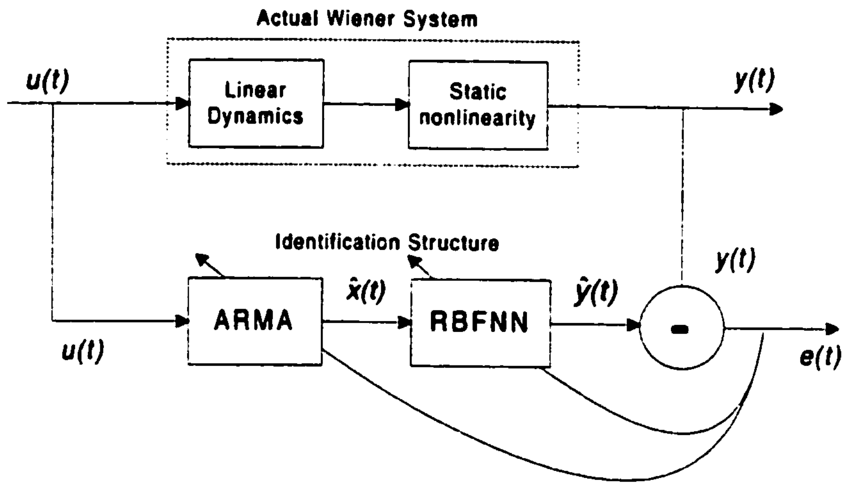


Figure 3.2: Identification of Wiener system.

The parameters (weights of RBFNN and the coefficients of the ARMA) are updated by minimizing the performance index $I$ given by,

$$I = \frac{1}{2}e^2(t).$$

where, $e(t) = y(t) - \hat{y}(t)$ and $\hat{y}(t)$ is the estimated output of the Wiener model.

The coefficients of the ARMA model and the weights of the RBFNN should be updated in the negative direction of the gradient. Keeping the coefficients in a parameter vector $\theta$ as $\theta = [a_1 \; \ldots \; a_n \;\; b_o \; \ldots \; b_m]$ and the regressions of the quantities $x(t)$ and $u(t)$ in regression vector $\psi(t)$ as $\psi(t) = [\hat{x}(t-1) \; \ldots \; \hat{x}(t-n) \; \hat{u}(t-d) \; \ldots \; \hat{u}(t-m-d)]$ and finding partial derivative. The derivation goes as follows:

$$
\begin{aligned}
\frac{\partial I}{\partial \theta} &= \frac{1}{2}\frac{\partial e^2(t)}{\partial \theta}, \\
&= e(t)\frac{\partial}{\partial \theta}\left(y(t) - \hat{y}(t)\right), \\
&= e(t)\frac{\partial}{\partial \theta}\left(y(t) - W\Phi(t)\right), \\
&= e(t)\frac{\partial}{\partial \theta}\left(y(t) - \sum_{i=1}^{n_o} w_i \phi(\|\hat{x}(t) - c_i\|)\right), \\
&= -e(t)\frac{\partial}{\partial \theta}(w_1 \phi(\|\hat{x}(t) - c_1\|) + \cdots + w_{n_o}\phi(\|\hat{x}(t) - c_{n_o}\|)), \\
&= -e(t)\frac{\partial}{\partial \theta}(w_1 \exp(-\frac{\|\hat{x}(t) - c_1\|^2}{\beta^2}) + \cdots + w_{n_o}\exp(-\frac{\|\hat{x}(t) - c_{n_o}\|^2}{\beta^2})), \\
&= -e(t)\frac{\partial}{\partial \theta}(w_1 \exp(-\frac{(\sqrt{(\hat{x}(t) - c_1)^2})^2}{\beta^2}) + \cdots + w_{n_o}\exp(-\frac{(\sqrt{(\hat{x}(t) - c_{n_o})^2})^2}{\beta^2})), \\
&= -e(t)\frac{\partial}{\partial \theta}(w_1 \exp(-\frac{(\hat{x}(t) - c_1)^2}{\beta^2}) + \cdots + w_{n_o}\exp(-\frac{(\hat{x}(t) - c_{n_o})^2}{\beta^2})), \\
&= -e(t)\frac{\partial}{\partial \theta}(w_1 \exp(-\frac{(\hat{x}(t) - c_1)^2}{\beta^2}) + \cdots + w_{n_o}\exp(-\frac{(\hat{x}(t) - c_{n_o})^2}{\beta^2})). \qquad (3.7)
\end{aligned}
$$

Let $\Omega_i = w_i \exp(-\frac{(\hat{x}(t) - c_i)^2}{\beta^2})$, where $i = [1, 2, \ldots n_o]$. Considering the partial derivative of $\Omega_j$ term $w.r.t.$ any $a_i$,

$$
\begin{aligned}
\frac{\partial\Omega_j}{\partial a_i} &= \frac{\partial}{\partial a_i} w_j \exp(-\frac{(\hat{x}(t)-c_j)^2}{\beta^2}), \\
&= -w_j \exp(-\frac{(\hat{x}(t)-c_j)^2}{\beta^2}) \frac{\partial}{\partial a_i} \frac{(\hat{x}(t)-c_j)^2}{\beta^2}, \\
&= -2(\hat{x}(t)-c_j)\frac{w_j}{\beta^2} \exp(-\frac{(\hat{x}(t)-c_j)^2}{\beta^2}) \frac{\partial}{\partial a_i} \frac{B(q^{-1})}{A(q^{-1})} q^{-d}\hat{u}(t), \\
&= -2(\hat{x}(t)-c_j)\frac{w_j}{\beta^2} \exp(-\frac{(\hat{x}(t)-c_j)^2}{\beta^2}) \frac{\partial}{\partial a_i} (\sum_{l=1}^{n} a_l \hat{x}(t-l) + \sum_{k=0}^{m} b_k \hat{u}(t-k-d)), \\
&= -2(\hat{x}(t)-c_j)\frac{w_j}{\beta^2} \exp(-\frac{(\hat{x}(t)-c_j)^2}{\beta^2}) \hat{x}(t-i), \\
\frac{\partial\Omega_j}{\partial a_i} &= -2\hat{x}(t-i)(\hat{x}(t)-c_j)\frac{w_j}{\beta^2} \exp(-\frac{(\hat{x}(t)-c_j)^2}{\beta^2}). \qquad (3.8)
\end{aligned}
$$

Putting the $\frac{\partial\Omega_j}{\partial a_i}$ term from Eq. 3.8 in Eq. 3.7 for $a_i$,

$$
\begin{aligned}
\frac{\partial I}{\partial a_i} &= -2e(t)\hat{x}(t-i)(\hat{x}(t)-c_1)\frac{w_1}{\beta^2} \exp(-\frac{(\hat{x}(t)-c_1)^2}{\beta^2}) \\
&\quad - 2e(t)\hat{x}(t-i)(\hat{x}(t)-c_2)\frac{w_2}{\beta^2} \exp(-\frac{(\hat{x}(t)-c_2)^2}{\beta^2}) - \dots \\
&\quad \dots - 2e(t)\hat{x}(t-i)(\hat{x}(t)-c_{n_o})\frac{w_{n_o}}{\beta^2} \exp(-\frac{(\hat{x}(t)-c_{n_o})^2}{\beta^2}).
\end{aligned}
$$

or, in closed form,

$$
\begin{aligned}
\frac{\partial I}{\partial a_i} &= -\frac{2e(t)}{\beta^2}\hat{x}(t-i)\sum_{j=1}^{n_o}(\hat{x}(t)-c_j)w_j \exp(-\frac{(\sqrt{(\hat{x}(t)-c_j)^2})^2}{\beta^2}). \\
\frac{\partial I}{\partial a_i} &= -\frac{2e(t)}{\beta^2}\hat{x}(t-i)\sum_{j=1}^{n_o}(\hat{x}(t)-c_j)w_j\phi(\|\hat{x}(t)-c_j\|).
\end{aligned}
$$

Similarly for any $b_i$,

$$\frac{\partial I}{\partial b_i} = -\frac{2e(t)}{\beta^2}\hat{u}(t-i-d)\sum_{j=1}^{n_o}(\hat{x}(t)-c_j)w_j\phi(\|\hat{x}(t)-c_j\|),$$

and finally stacking the derivatives in Eq. 3.7 again,

$$\begin{aligned}\frac{\partial I}{\partial \theta} &= -e(t)\Big(-\frac{2}{\beta^2}\psi(t)\sum_{j=1}^{n_o}(\hat{x}(t)-c_j)w_j\phi(\|\hat{x}(t)-c_j\|)\Big),\\ \frac{\partial I}{\partial \theta} &= \frac{2e(t)}{\beta^2}\psi(t)\sum_{j=1}^{n_o}(\hat{x}(t)-c_j)w_j\phi(\|\hat{x}(t)-c_j\|).\end{aligned}$$

Now, this gradient is used in finding the updated parameters using Eq. 3.2.

$$\boxed{\theta(K+1) = \theta(K) - \frac{2\alpha e(t)}{\beta^2}\psi(t)\sum_{j=1}^{n_o}(\hat{x}(t)-c_j)w_j\phi(\|\hat{x}(t)-c_j\|).} \tag{3.9}$$

The partial derivatives for the weights are derived as follows,

$$\begin{aligned}\frac{\partial I}{\partial W} &= \frac{1}{2}\frac{\partial e^2(t)}{\partial W},\\ &= e(t)\frac{\partial}{\partial W}\left(y(t)-\hat{y}(t)\right),\\ &= e(t)\frac{\partial}{\partial W}\left(y(t)-W\Phi(t)\right).\\ \frac{\partial I}{\partial W} &= -e(t)\Phi(t).\end{aligned} \tag{3.10}$$

The gradient in Eq. 3.10 is used to find the updated weights using Eq. 3.3. The final weight update equation will take the form,

$$\boxed{W(K+1) = W(K) + \alpha\, e(t)\Phi(t).} \tag{3.11}$$

# Chapter 4

# Simulation Results for SISO Systems

This section comprises of the validation of the proposed training algorithm for the identification of the SISO Hammerstein and Wiener models. The proposed training algorithms are applied on several practical examples for both Hammerstein and Wiener models including heat exchanger [21]. control valve [23] and saturation non-linearity [69].

To inspect the effect of noise on the accuracy of identification scheme, simulations are also performed in noisy environment for both Hammerstein and Wiener models.

## 4.1 Simulation Results for SISO Hammerstein Model

### Example 1: Saturation Nonlinearity

The process used in this example is a second-order system with a saturation nonlinearity at the input. The plant is given by the following difference equation:

$$y(t) = 0.4y(t-1) + 0.35y(t-2) + 0.8x(t), \tag{4.1}$$

where $x(t)$ is a nonlinear function of the input $u(t)$ and its characteristics are given by,

$$x(t) = \begin{cases} 0.5, & \text{for } u(t) > 0.5 \\ u(t), & \text{for } -0.5 \leq u(t) \leq 0.5 \\ -0.5, & \text{for } u(t) < -0.5 \end{cases} \tag{4.2}$$

Using random inputs $u(t)$ uniformly distributed in the interval $[-2, 2]$, the desired outputs are generated by means of the process model given by Eq. 4.1.

The structure of the identification model comprised of an RBFNN in series with an ARMA model. The centers of the RBFNN are evenly distributed in input space.

The ARMA model employed is given by:

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + b_o x(t). \tag{4.3}$$

For this example, after few trials, the width of the basis functions and the learning rate are optimized to 0.41 and 0.04, respectively.

The proposed algorithm is applied to update the weights of the RBFNN and the coefficients of the ARMA. The RBFNN identified the nonlinearity in the model. The actual and identified saturation nonlinearities shown in Fig. 4.2, reflect excellent approximation. The square error plot is shown in Fig. 4.3, and it depicts that the square error is minimized to 0.1 after 30 iterations. The estimated parameters $\hat{a}_1$, $\hat{a}_2$ and $\hat{b}_o$ are converged to values 0.6026, 0.3521 and 0.6889, respectively. The behaviour of parameter convergence is shown in Fig. 4.4.

The values of the coefficients are very close to the true values, except for the $b_o$. To study the variation in the value of $b_o$, it must be made clear that $b_o$ can be considered as a linear gain that can be factored out from the linear subsystem as shown in Fig. 4.1.

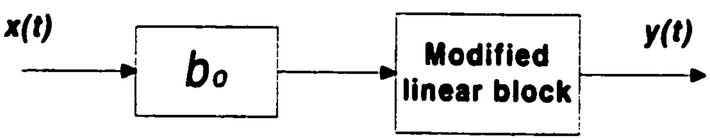


Figure 4.1: Modified linear block in the Hammerstein model with $b_o$ factored out.

The modified linear block can be written as,

$$y(t) = \frac{1 + \frac{b_1}{b_o}q^{-1} + \frac{b_2}{b_o}q^{-2} + \cdots + \frac{b_n}{b_o}q^{-n}}{1 + a_1q^{-1} + a_2q^{-2} + \cdots + a_mq^{-m}} b_o x(t). \tag{4.4}$$

This implies that $b_o$ appears as a product term with $x(t)$, *i.e.* the output of the nonlinearity, so the intermediate variable is $\hat{b}_o \times \hat{x}(t)$. It is in effect evident that the values constituting a product cannot be distinguished out from the product itself. Therefore, the identification in the simulations are done over the actual $b_o \times x(t)$ and the estimated $\hat{b}_o \times \hat{x}(t)$. The rest of the parameters of the numerator *i.e.*, $b_1$, $b_2, \ldots, b_n$ are also identified as a ratio between the estimate of the parameter itself and $\hat{b}_o$. Similarly, the nonlinearities are also plotted as a product of actual $b_o$ and the output of the actual nonlinearity; and, the product of the estimated $\hat{b}_o$ and the output of the estimated nonlinearity, .

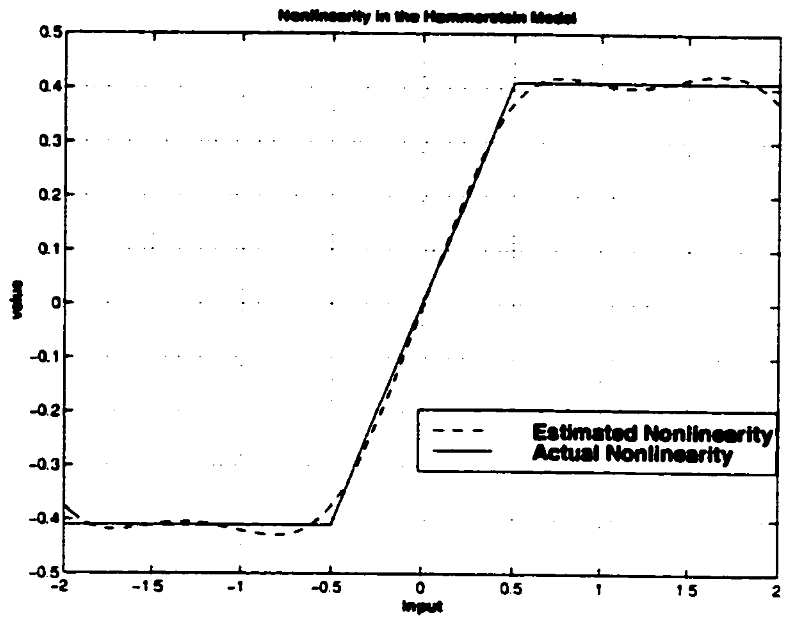


Figure 4.2: Actual and identified saturation nonlinearities for example 1 of Hammerstein model.

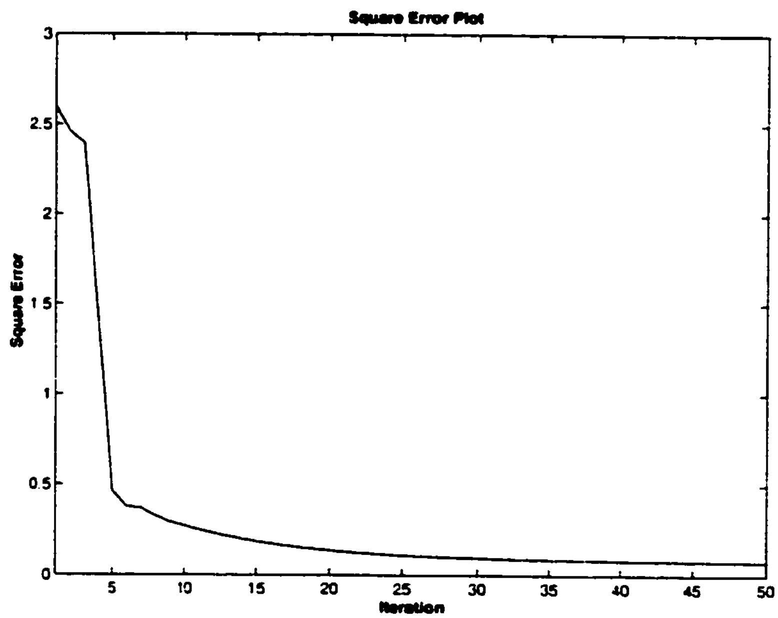


Figure 4.3: Square error plot for example 1 of Hammerstein model.

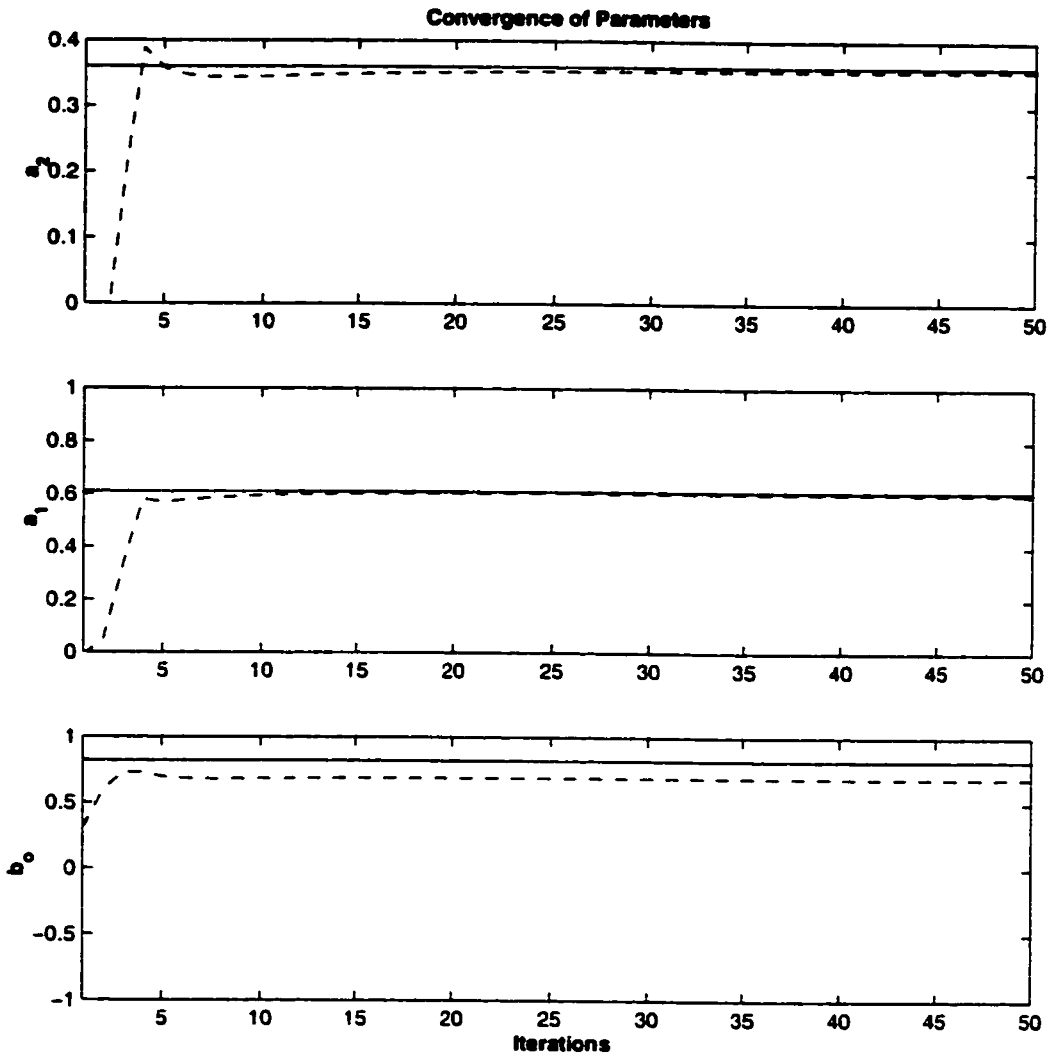


Figure 4.4: Convergence of parameters for example 1 of Hammerstein model.

## Example 2: Exponential Cosine

In this example, a third order system is considered as

$$y(t) = 0.4y(t-1) + 0.35y(t-2) + 0.1y(t-3) + 0.8x(t) - 0.2x(t-1). \qquad (4.5)$$

The static nonlinearity $x(t)$ is an exponential cosine function of the input $u(t)$ and is expressed as,

$$x(t) = \cos(3u(t)) + \exp(-|u(t)|). \qquad (4.6)$$

The linear part is modelled by a third order ARMA.

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + a_3 y(t-3) + b_o x(t) + b_1 x(t-1).$$

The input signal used for the simulation is a set of uniformly distributed random number in the range $[-2, 2]$. The desired output are generated using the process model given by Eq. 4.5. The centers of the basis functions are located evenly in the input space.

The width of the basis function is set to 0.5 and the learning rate 0.04. These values are selected after few trial runs.

The developed algorithm is employed to update the weights of RBFNN and coefficients of ARMA. The RBFNN identified the nonlinearity and the ARMA identified the linear part. The actual and identified nonlinearities shown in Fig. 4.5, show the accuracy of the identification algorithm. The square error plot is shown in Fig. 4.6, where the square error is minimized to 0.1 after approximately 20 iterations. The parameters $a_1$, $a_2$, $a_3$, $b_o$ and $b_1$ are converged to 0.4008, 0.3492, 0.1024, 0.6219 and $-0.2162$, respectively. The convergence of parameters is shown in Fig. 4.7. It is noted that the value of $b_o$ is different from the true value. The variation in the value of $b_o$ is discussed in example 1.

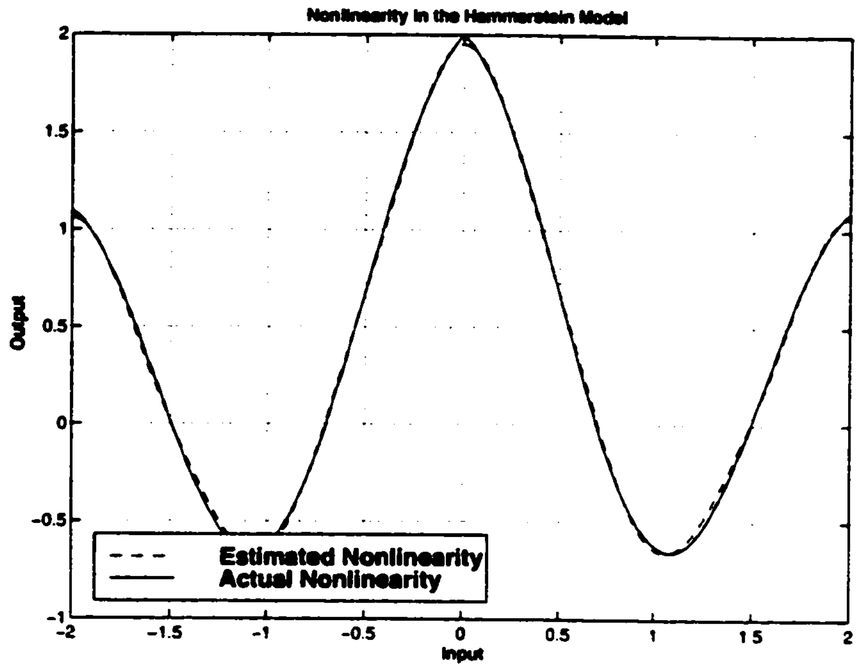


Figure 4.5: Actual and identified exponential cosine nonlinearity for example 2 of Hammerstein model.

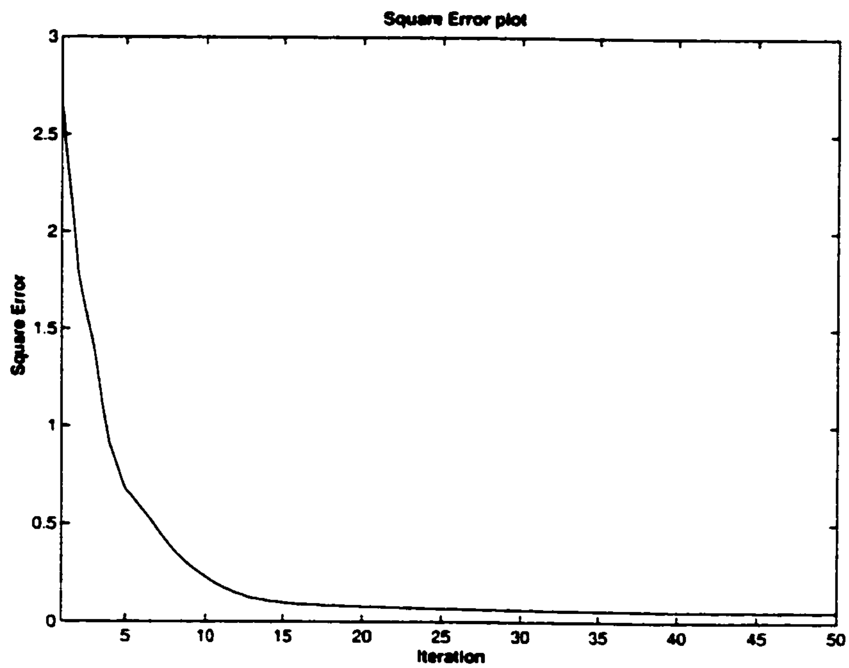


Figure 4.6: Square error plot for example 2 of Hammerstein model.

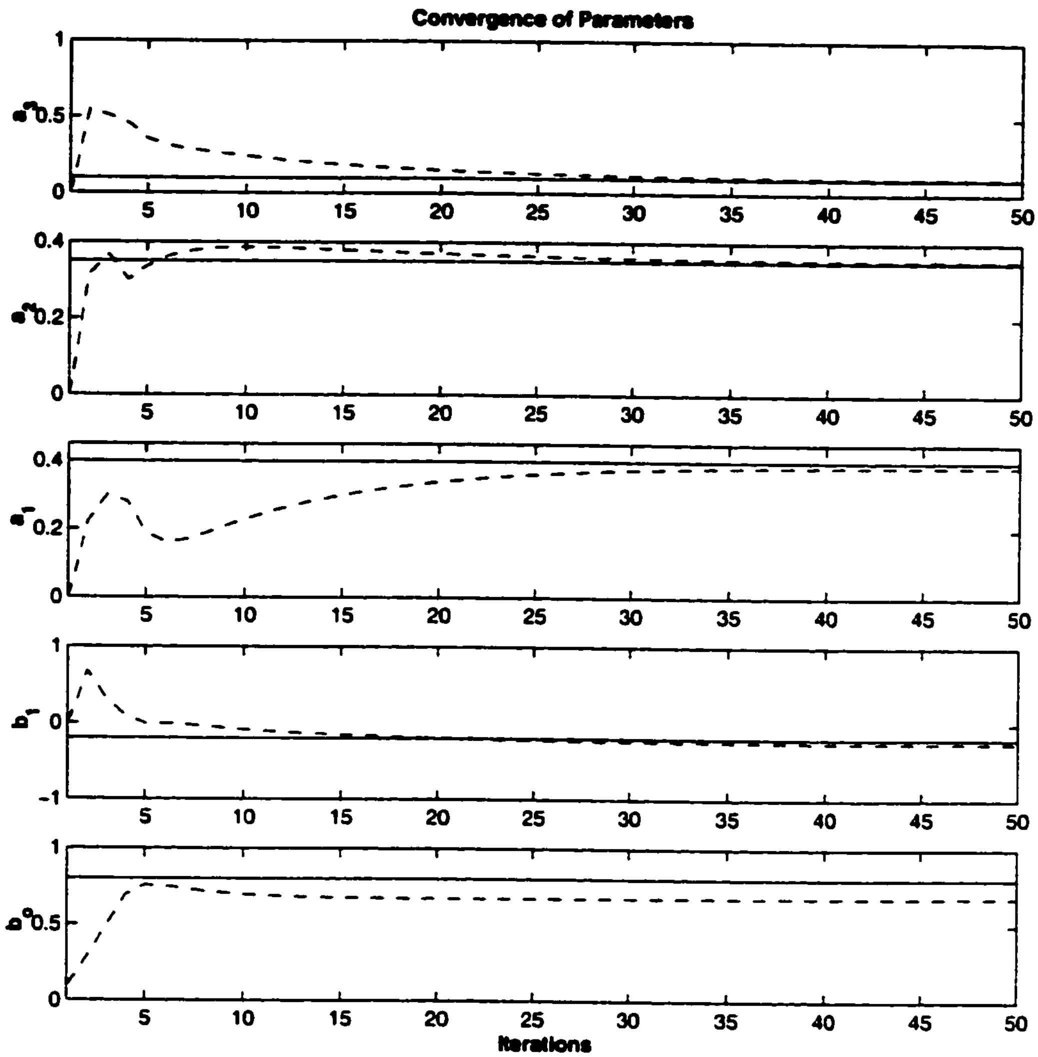


Figure 4.7: Convergence of parameters for example 2 of Hammerstein model.

## Example 3: Heat Exchanger

In this example, a heat exchanger, which is modeled as a Hammerstein model [24], is considered with the same linear third order dynamics as used in example 2 given by,

$$y(t) = 0.4y(t-1) + 0.35y(t-2) + 0.1y(t-3) + 0.8x(t) - 0.2x(t-1). \tag{4.7}$$

where $x(t)$ is a function of input $u(t)$, given by,

$$x(t) = -31.549u(t) + 41.732u^2(t) - 24.201u^3(t) + 68.634u^4(t). \tag{4.8}$$

The input signal is a set of uniformly distributed random number in the range $[-2, 2]$. These inputs are employed to produce the desired outputs using the process model given by Eq. 4.7. The identification structure is composed of an RBFNN in series with an ARMA model. The centers of the basis functions are located in the input space. After carrying out few trials, the width of the basis function is set to 0.6 and the learning rate is set to 0.05. The ARMA model was given by the following difference equation:

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + a_3 y(t-3) + b_o x(t) + b_1 x(t-1).$$

The identification algorithm is implemented to identify the nonlinearity and the coefficients of the linear part. The actual and identified nonlinearities for the given (normalized) heat exchanger are shown in Fig. 4.8, reflecting impressive identification behaviour. The square error plot is shown in Fig. 4.9, where a value of 0.22 is achieved after approximately 20 iterations. The parameters $a_1$, $a_2$, $a_3$, $b_o$ and $b_1$ are converged to 0.3989, 0.3519, 0.1058, 0.5275 and −0.1994, respectively. The convergence of parameters are shown in Fig. 4.10. Similar to example 1 and 2, the value of $b_o$ is not accurately estimated but the variation in the estimation in discussed in example 1 in detail.

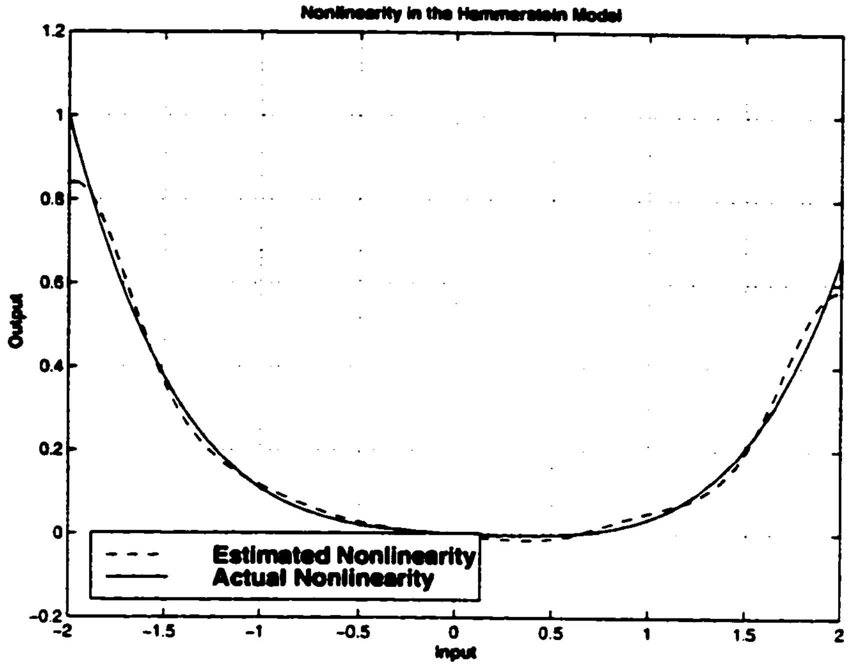


Figure 4.8: Actual and identified heat exchanger for example 3 of Hammerstein model.

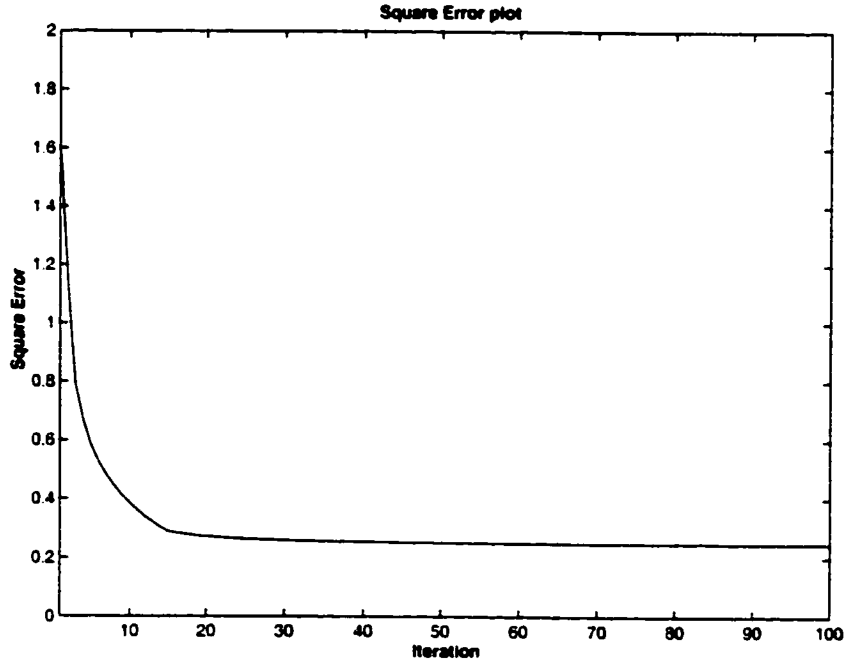


Figure 4.9: Square error plot for example 3 of Hammerstein model.

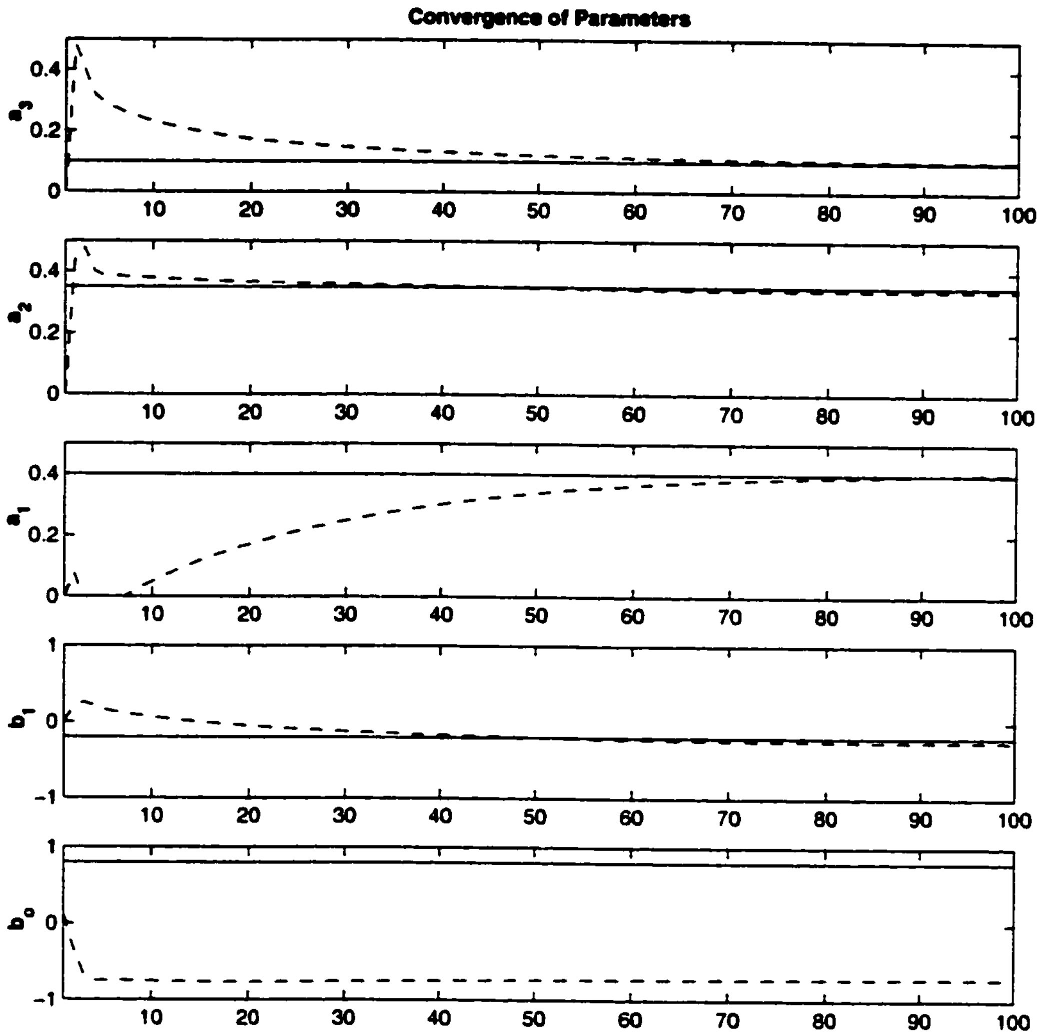


Figure 4.10: Convergence of parameters for example 3 of Hammerstein model.

## Example 4: Heat Exchanger in a Noisy Environment

To study the effect of noise on the accuracy of the identification scheme, the heat exchanger example is considered in noisy environment. The disturbance considered in this example is zero mean white Gaussian noise. The noise $n(t)$ is additive in nature and adding at the output $y(t)$ of the Hammerstein model. The Hammerstein model in the noisy environment is shown in Fig. 4.11, where $y_n(t)$ is the available noisy output.

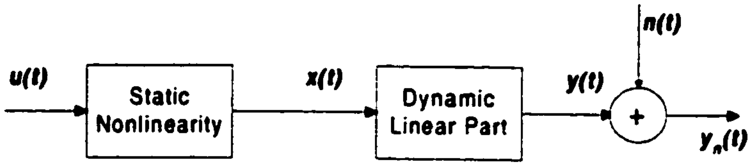


Figure 4.11: Hammerstein model with output additive noise.

Two cases with different noise powers are considered, such that the signal to noise ratios (SNRs) are 30 dB and 20 dB. The process model is the same as considered in example 3 with the static nonlinearity at the input given by.

$$x(t) = -31.549u(t) + 41.732u^2(t) - 24.201u^3(t) + 68.634u^4(t). \qquad (4.9)$$

The linear dynamics of the model are given by the following difference equation.

$$y(t) = 0.4y(t-1) + 0.35y(t-2) + 0.1y(t-3) + 0.8x(t) - 0.2x(t-1). \qquad (4.10)$$

The measured or observed output is given by.

$$y_n(t) = y(t) + n(t) \qquad (4.11)$$

Using random inputs $u(t)$ uniformly distributed in the interval $[-2, 2]$, the outputs are generated by means of the process model given by Eqs. 4.9, 4.10 and 4.11.

The identification structure composed of an RBFNN in series with an ARMA. The basis functions are centered evenly in the range $[-2, 2]$. The width of the basis function and the learning rate is selected as 0.6 and 0.05, respectively, after few trial runs. The ARMA model used in the identification is of the following form:

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + a_3 y(t-3) + b_o x(t) + b_1 x(t-1).$$

The proposed algorithm is applied to update the weights of the RBFNN and coefficients of the ARMA. The nonlinearities in both the cases are identified shown in Fig. 4.1 and Fig. 4.1, revealing the competence and reliability of the identification scheme even in the noisy environment. The parameters of the linear part are also converged to values very close to the true values except for $b_o$ due to the reason explained in example 1. The estimates of parameters are shown in table 4.1.

Table 4.1: Parameter estimates of Hammerstein model in noisy environment.

| | $b_o$ | $a_1$ | $a_2$ |
|---|---|---|---|
| True value | 1.8 | 0.51 | -0.35 |
| SNR | $\hat{b}_o$ | $\hat{a}_1$ | $\hat{a}_2$ |
| 30dB | -1.319 | 0.491 | -0.343 |
| 20dB | -1.104 | 0.470 | -0.324 |

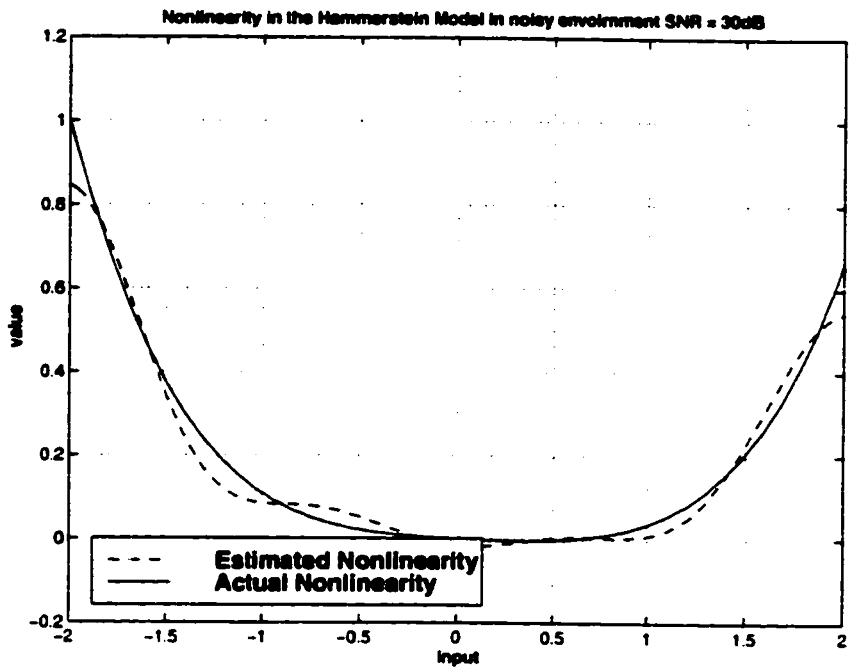


Figure 4.12: Heat exchanger identification in Hammerstein model with SNR 30dB

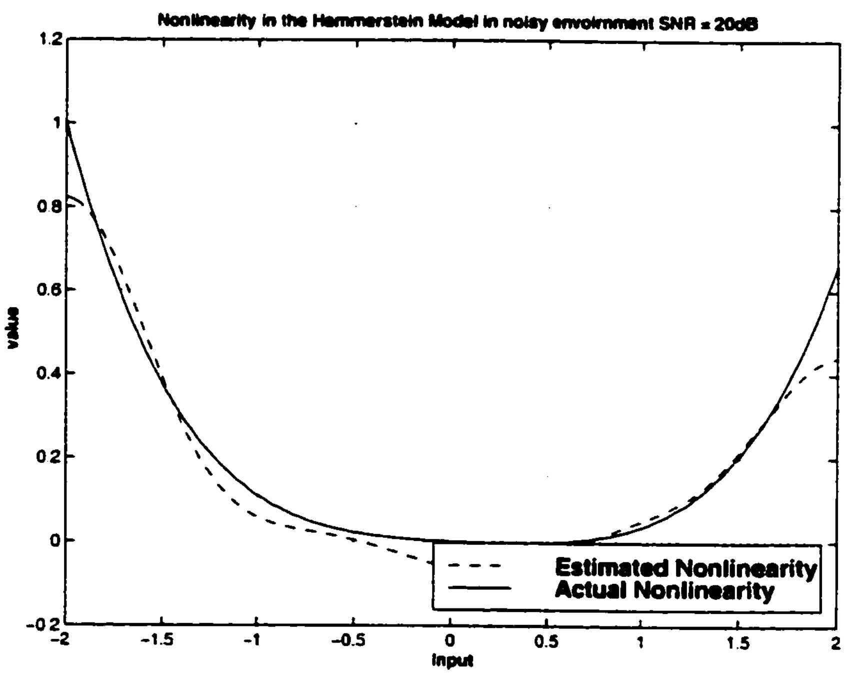


Figure 4.13: Actual and identified heat exchanger in Hammerstein model with SNR 20dB

## 4.2 Simulation Results for SISO Wiener Model

### Example 1: Saturation Nonlinearity

The process used in this example is a second-order system with a saturation nonlinearity at the output. The characteristics of the saturation nonlinearity are given by,

$$y(t) = \begin{cases} 0.5, & for \;\; x(t) > 0.5 \\ x(t), & for \;\; -0.5 \leq x(t) \leq 0.5 \\ -0.5. & for \;\; x(t) < -0.5. \end{cases} \tag{4.12}$$

The linear part is taken as:

$$x(t) = 0.4x(t-1) - 0.55x(t-2) + 0.8u(t). \tag{4.13}$$

The structure of the identification model comprised of an ARMA model in series with an RBFNN. The linear part is modeled by the ARMA model described by.

$$x(t) = a_1 x(t-1) + a_2 x(t-2) + b_o u(t),$$

The nonlinearity is identified by the RBFNN. The centers of the RBFNN are located evenly in the range $[-2, 2]$. The width and the learning rate are selected after few trial runs as, 0.41 and 0.5, respectively. Using random inputs $u(t)$ uniformly distributed in the interval $[-2, 2]$, the desired outputs are generated by employing the process model given by Eq. 4.12 and Eq. 4.13.

The proposed learning algorithm is used to update the weights of the RBFNN and the parameters of the ARMA model. The actual and identified nonlinearity of the identified model after training are shown in Fig. 4.15 and reveal the accuracy of the identification algorithm. The square error plot is sown in Fig. 4.16 where the square error is minimized to a value less than 0.1 after 80 iterations and achieved steady state value after 140 iterations.

The parameters $a_1$, $a_2$ and $b_o$ are converged to values 0.3994. -0.5512 and 0.7316, respectively. The convergence of parameters is shown in Fig. 4.17. From the estimated values and the Fig. 4.17 it is clear that $b_o$ is not converged to its true value. This phenomenon was also observed in the Hammerstein model simulation examples. Similar to Hammerstein model, the $b_o$ can be factored out of the linear block, as shown in Fig. 4.14.

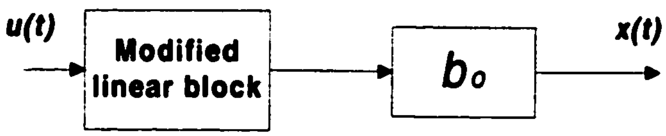


Figure 4.14: Modified linear block in the Wiener model with $b_o$ factored out.

The modified linear block can be written as,

$$x(t) = \frac{1 + \frac{b_1}{b_o}q^{-1} + \frac{b_2}{b_o}q^{-2} + \cdots + \frac{b_n}{b_o}q^{-n}}{1 + a_1q^{-1} + a_2q^{-2} + \cdots + a_mq^{-m}} b_o u(t), \tag{4.14}$$

$$\text{or} \quad \frac{x(t)}{b_o} = \frac{1 + \frac{b_1}{b_o}q^{-1} + \frac{b_2}{b_o}q^{-2} + \cdots + \frac{b_n}{b_o}q^{-n}}{1 + a_1q^{-1} + a_2q^{-2} + \cdots + a_mq^{-m}} u(t). \tag{4.15}$$

This means that $b_o$ is with $x(t)$, *i.e.* the output of the nonlinearity, so the intermediate variable is $\frac{\hat{x}(t)}{\hat{b}_o}$. It is commonly known that the values constituting a fraction cannot be distinguished from the quotient of the fraction itself. Therefore, the identification in the simulations are done over the actual $\frac{x(t)}{b_o}$; the estimated $\frac{\hat{x}(t)}{\hat{b}_o}$, each of the $x(t)$ and $b_o$ cannot be solely identified. Similarly, the parameters $b_1, b_2, \ldots b_n$ are also identified as a ratio between the estimated parameter itself and $\hat{b}_o$.

In the identification of the static nonlinearity, it is observed that the magnitude of the estimated $\hat{b}_o$ did not effect the nonlinearity, but had an effect on its polarity. This means that estimated nonlinearity is the same as the actual one if the sign of $\hat{b}_o$ is taken into consideration, *i.e.* if the input to the nonlinearity is fed along with the sign of $\hat{b}_o$, the resulting nonlinearity approximates the actual nonlinearity.

In the Hammerstein model, the nonlinearity is at the input. Therefore, the input signals to the actual nonlinearity and RBFNN are same, during both the training and testing phases. As discussed above, in the case of identification of Wiener systems, the intermediate variable $x(t)$ is not the same for the actual and identified model.

This implies that the input to the actual nonlinearity and the RBFNN differs during training, *i.e.*, the weights of the RBFNN are updated for some different input-output pairs as compared to the actual system. The estimated $\hat{b}_o$ is dissimilar to the actual $b_o$ and can also be the negative of the actual $b_o$, inverting the phase of the input to the RBFNN.

Now, during the testing phase, where a test input is used to identify and compare the static nonlinearity entirely separate from the linear part, the inputs supplied to both the actual nonlinearity and the RBFNN are identical. The only concern is that the RBFNN is possibly trained with a phase inverted signal due to the sign of $\hat{b}_o$. If the sign of $\hat{b}_o$ is the same as $b_o$, then identified static nonlinearity approximates the actual and if the sign of $\hat{b}_o$ is opposite to the actual $b_o$ the RBFNN provides an estimate that is a negative of the actual nonlinearity.

Keeping the sign of $\hat{b}_o$ with the input during the testing phase makes sure that the RBFNN is truly identifying the actual nonlinearity in the system.

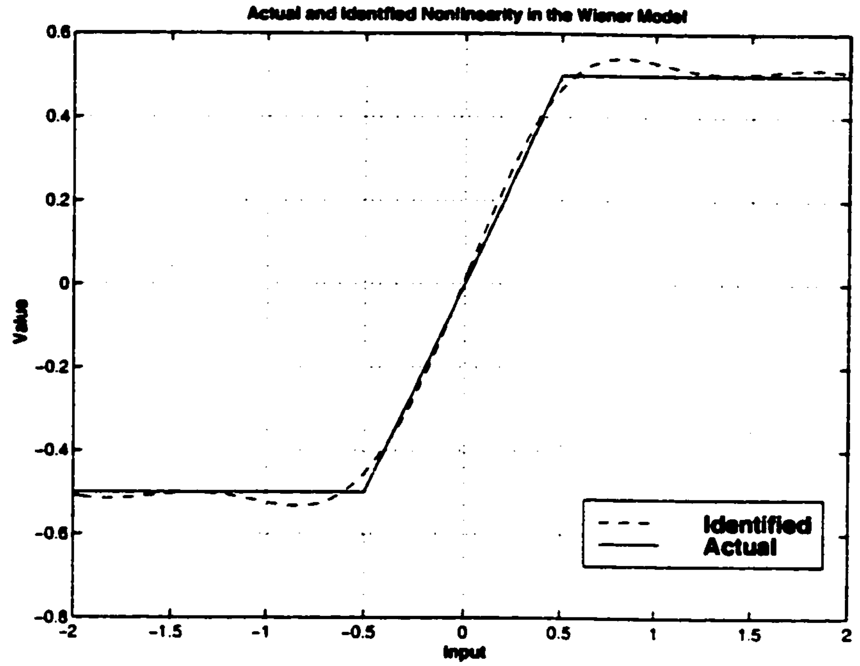


Figure 4.15: Actual and identified saturation nonlinearities for example 1 of Wiener model.

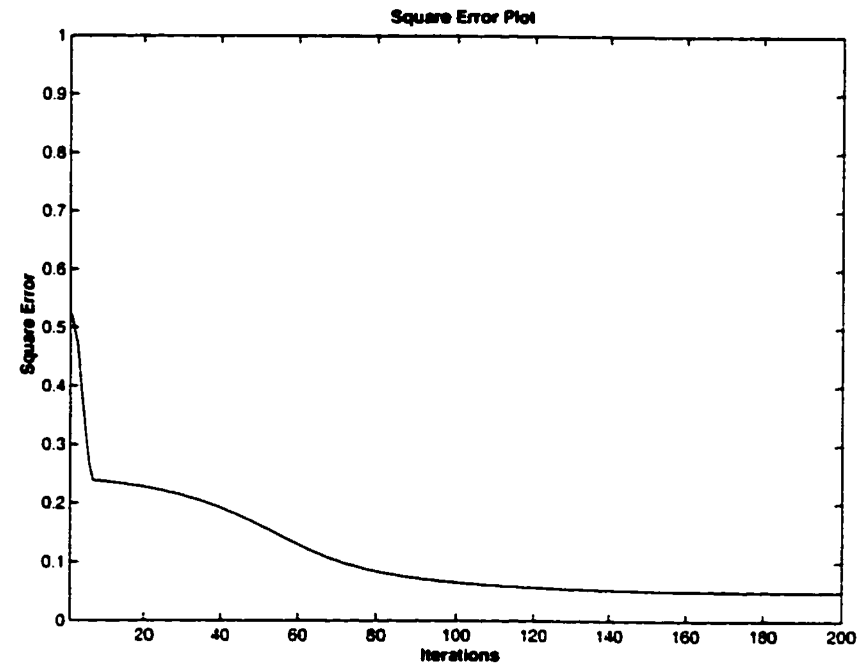


Figure 4.16: Square error plot for example 1 of Wiener model.

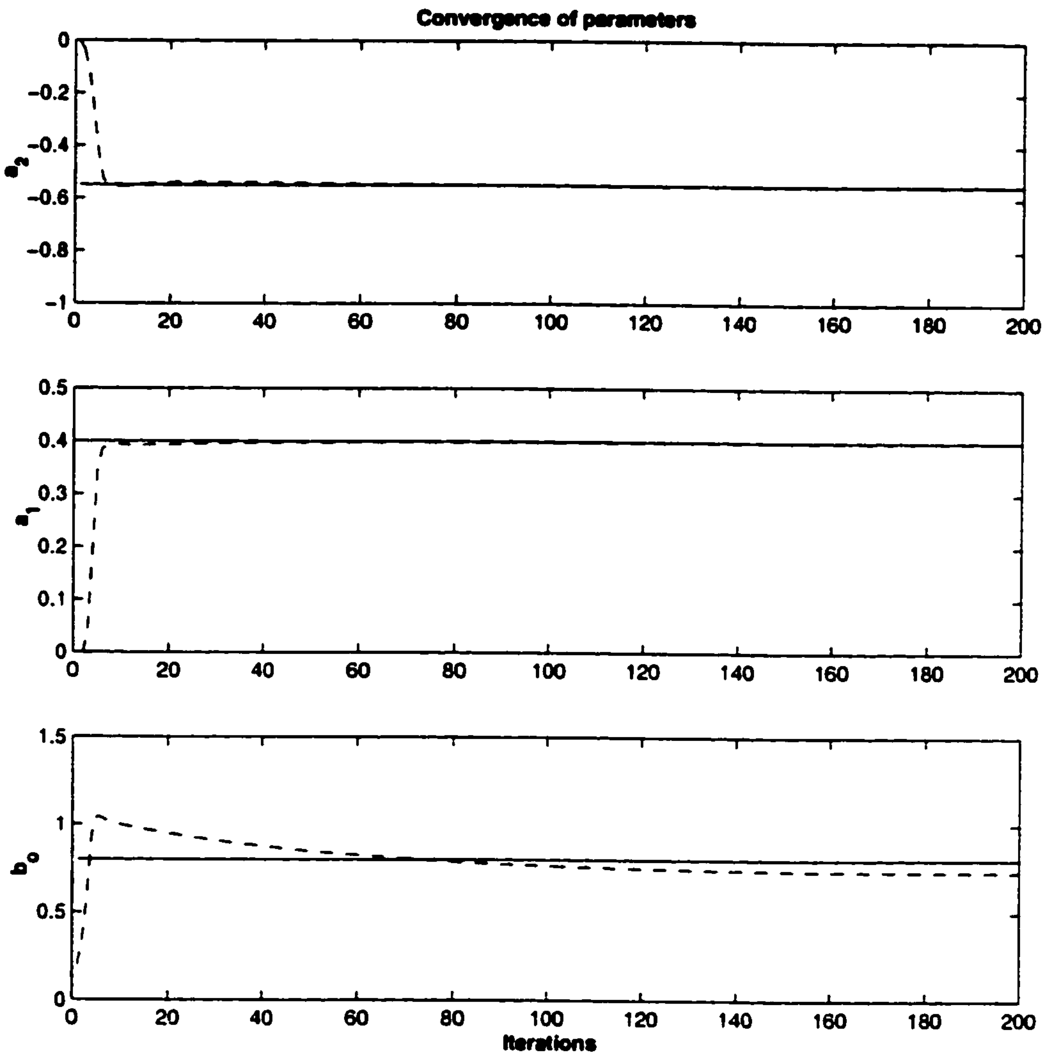


Figure 4.17: Convergence of parameters for example 1 of Wiener model.

## Example 2: Exponential Cosine

In this example, a third order linear system with an exponential cosine static nonlinearity at the output is considered. The linear part is given by,

$$x(t) = 0.4x(t-1) + 0.35x_2(t-2) + 0.1x_3(t-3) + 0.8u(t) - 0.2b_1u(t-1). \quad (4.16)$$

The exponential cosine nonlinear function is expressed as,

$$y(t) = \cos(3x(t)) + \exp(-|x(t)|). \quad (4.17)$$

The identification structure consisted of an RBFNN in series with an ARMA model. The ARMA model is given by,

$$x(t) = a_1x(t-1) + a_2x_2(t-2) + a_3x_3(t-3) + b_ou(t) + b_1u(t-1).$$

The centers of the basis functions are located evenly in the range $[-2, 2]$. The width of the basis function is kept as 0.51 and the learning rate is set to 0.04. These values are chosen after few trial runs.

The training data set is generated using random variables uniformly distributed in the interval $[-2, 2]$. The desired set of outputs is generated using Eq. 4.16 and Eq. 4.17.

The identification algorithm is applied to update the coefficients of the ARMA and the weights of the RBFNN. The RBFNN identified the nonlinearity accurately as shown in Fig. 4.18, where the identified nonlinearity overlaps the actual one. The ARMA model estimated the linear part and the parameters $a_1$, $a_2$, $a_3$, $b_o$ and $b_1$ are converged to values 0.4097, 0.3548, 0.0989, 0.7849 and -0.2179, respectively. The parameter $b_o$ did not converged to its true value, but is compensated with the static nonlinearity as discussed in example 1 of Wiener model. The convergence of parameters is shown in Fig. 4.20. The square error is minimized to a value 0.2 after approximately 50 iterations. The square error plot is shown in Fig. 4.19.

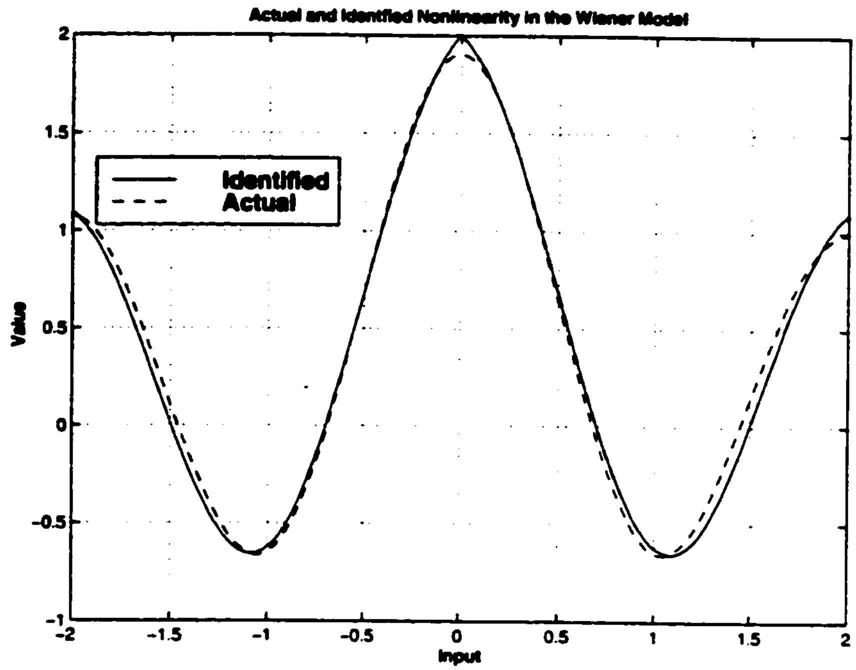


Figure 4.18: Actual and identified exponential cosine nonlinearities for example 2 of Wiener model.

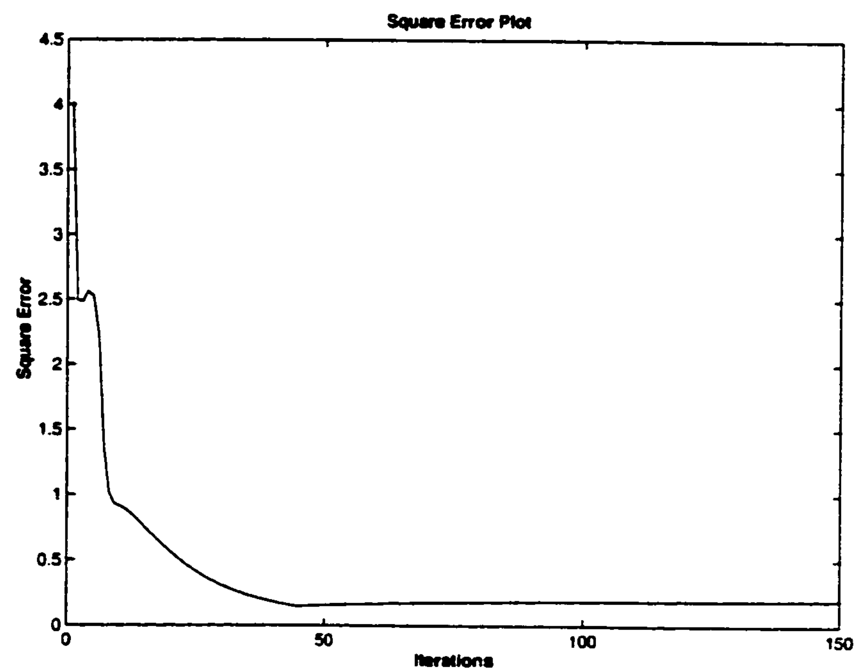


Figure 4.19: Square error plot for example 2 of Wiener model.

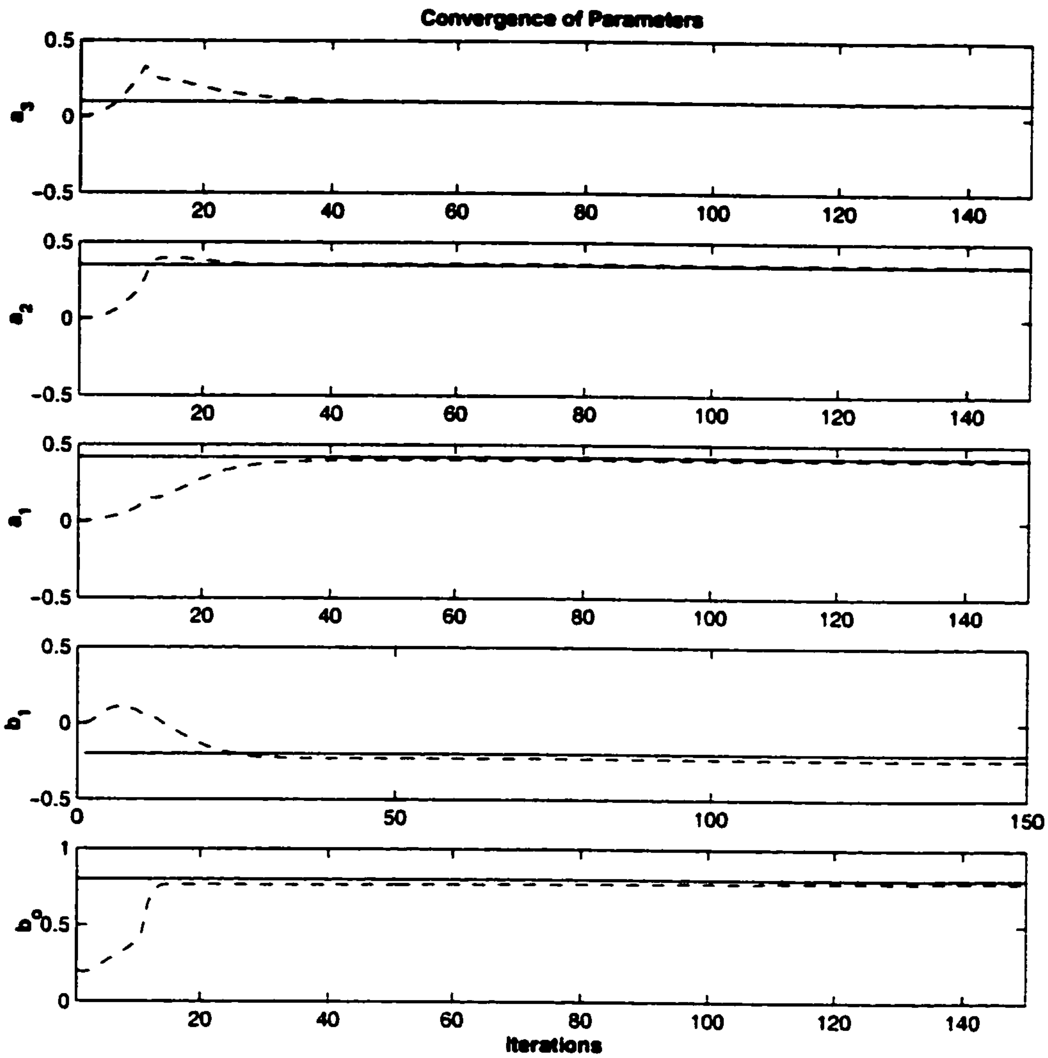


Figure 4.20: Convergence of parameters for example 2 of Wiener model.

## Example 3: Control Valve

In this example, the proposed identification algorithm is applied to a model that describes a valve for control of fluid flow described in [29] and [70]. The linear part is described by,

$$x(t) = 0.4x(t-1) + 0.35x_2(t-2) + 0.1x_3(t-3) + 0.8u(t) - 0.2b_1u(t-1). \quad (4.18)$$

and the nonlinear part is given by,

$$y(t) = \frac{x(t)}{\sqrt{0.10 + 0.90x^2(t)}}. \quad (4.19)$$

In this model, $u(t)$represents the pneumatic control signal applied to the stem of the valve and $x(t)$ represents the the stem position. The linear dynamics describe the dynamic balance between the control signal, a counteractive spring force and friction. The resulting flow through the valve is given by the nonlinear function of the stem position $x(t)$ reflected by $y(t)$.

The proposed identification algorithm is applied to estimate the linear and nonlinear parts of the model. To identify the linear part, an ARMA model structure was used given by,

$$x(t) = a_1x(t-1) + a_2x_2(t-2) + a_3x_3(t-3) + b_ou(t) + b_1u(t-1).$$

The nonlinearity is modeled by an RBFNN centered at $[-0.2, -0.1, 0.1, 2]$. The width of the basis functions and the learning rate are set to 1.1 and 0.04. The centers, the width of the basis functions and the learning rate are selected after few trial runs.

Using random inputs $u(t)$ uniformly distributed in the interval $[-2, 2]$, the desired outputs are generated by employing the process model given by Eq. 4.18 and Eq. 4.19.

The proposed identification algorithm is applied to update the weights of the RBFNN and the coefficients of the ARMA. The RBFNN identified the static nonlinearity accurately. The actual and identified nonlinearity of the identified model is shown in Fig. 4.21. The ARMA model estimated the linear part. The parameters $a_1$, $a_2$, $a_3$, $b_o$ and $b_1$ converged to values 0.4161, 0. 3491, 0.0992, 0.7082 and -0.2026, respectively. Similar to previous examples, $b_o$ is not converging to its true value, but it is compensated in the static nonlinearity. The convergence of parameters is shown in Fig. 4.23. The square error plot is shown in Fig. 4.22, where it is evident that the square error is minimized to a value 0.02 after 100 iterations.

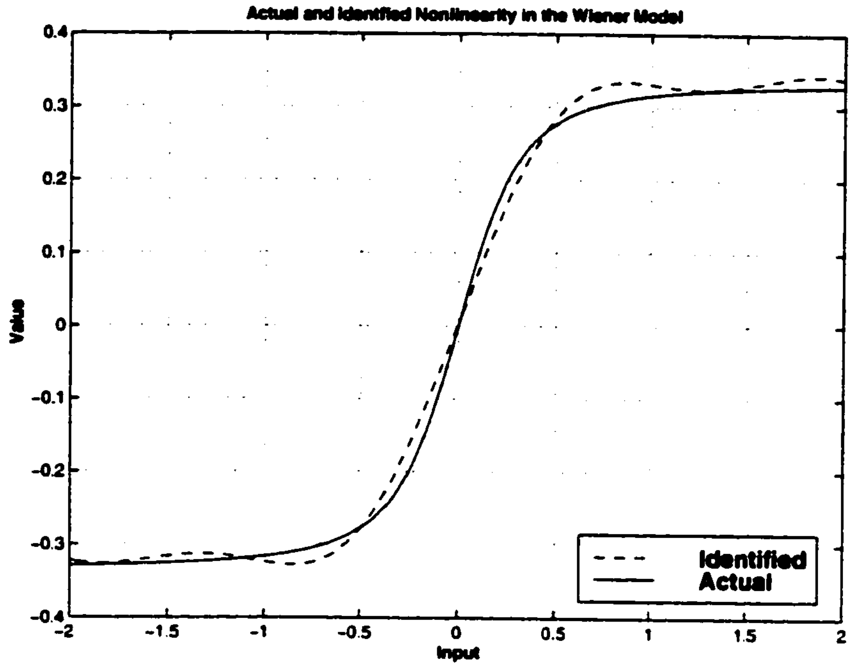


Figure 4.21: Actual and identified control valve for example 3.

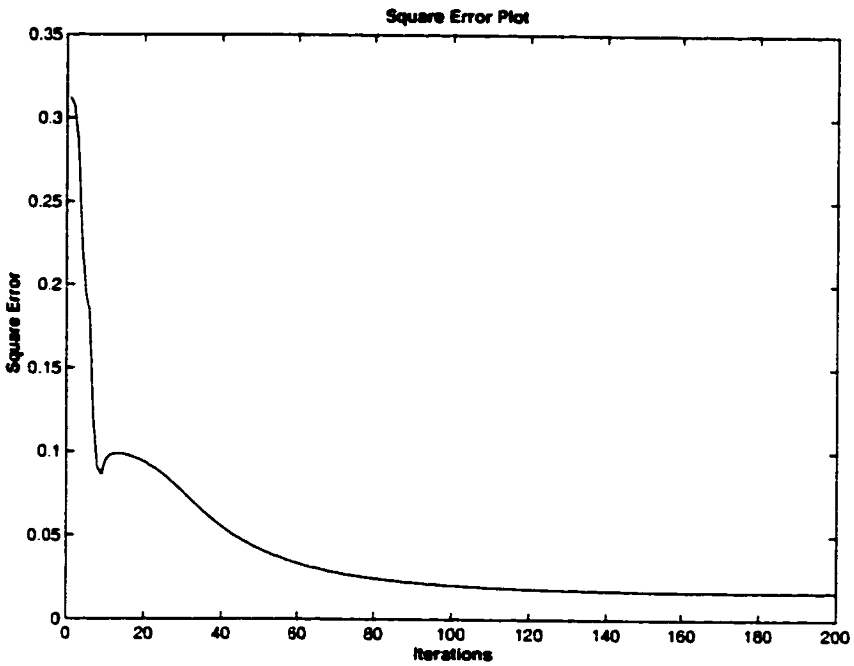


Figure 4.22: Square error plot for example 3 of Wiener model.

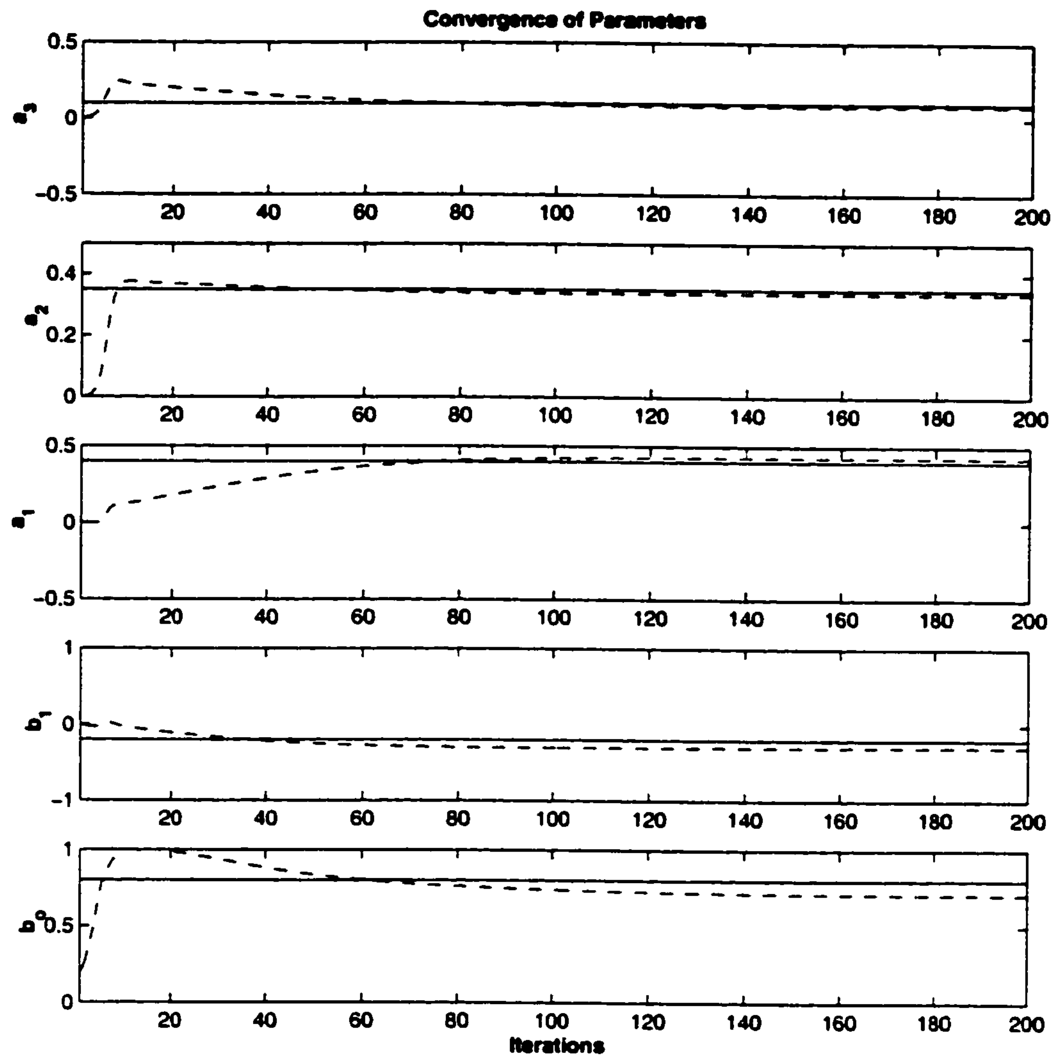


Figure 4.23: Convergence of parameters for example 3 of Wiener model.

## Example 4: Control Valve in Noisy Environment

To study the effect of noise on the accuracy of the identification scheme, the control valve example for the Wiener model is considered in noisy environment. The disturbance considered in this example is zero mean white Gaussian noise. The noise $n(t)$ is additive in nature and adding at the output $y(t)$ of the Wiener model. The Wiener model in the noisy environment is shown in Fig. 4.24, where $y_n(t)$ is the available noisy output.

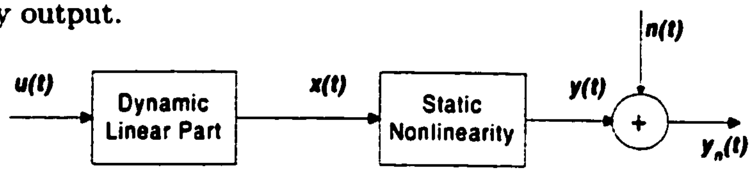


Figure 4.24: Wiener model with output additive noise.

Two cases with 30 dB SNR and 20 dB SNR are considered. The same process model as in example 3 of Wiener model is taken. The linear dynamics are given by,

$$x(t) = 0.4x(t-1) + 0.35x_2(t-2) + 0.1x_3(t-3) + 0.8u(t) - 0.2b_1u(t-1). \quad (4.20)$$

and the static nonlinearity is given by,

$$y(t) = \frac{x(t)}{\sqrt{0.10 + 0.90x^2(t)}}. \quad (4.21)$$

The measured or observed output is ,

$$y_n(t) = y(t) + n(t) \quad (4.22)$$

Using random inputs $u(t)$ uniformly distributed in the interval $[-2, 2]$, the outputs are generated by means of the process model given by Eq. 4.20, 4.21 and 4.22.

The identification structure is composed of an ARMA in series with an RBFNN. The basis functions are centered at $[-2, -0.1, 0.1, 2]$. The width of the basis function and the learning rate are selected as 1.1 and 0.04, respectively. The centers, the width and the learning rate are adjusted after few trial runs. The ARMA model used in the identification is of the following form:

$$y(t) = a_1y(t-1) + a_2y(t-2) + a_3y(t-3) + b_ox(t) + b_1x(t-1).$$

The proposed algorithm is used to update the weights of the RBFNN and coefficients of the ARMA. The nonlinearities in both the cases are identified with reasonable accuracy and are shown in Fig. 4.25 and Fig. 4.26. The parameters are also converged to their true values, that reveal the accuracy of the proposed scheme even in the noisy environment. The estimates of parameters are shown in table 4.2. The estimate of $b_o$ is not true for the reason discussed in example 1 of Wiener model.

Table 4.2: Parameter estimates for Wiener model in noisy environment.

| | $b_o$ | $a_1$ | $a_2$ |
|---|---|---|---|
| True value | 1.5 | 0.40 | -0.55 |
| SNR | $\hat{b}_o$ | $\hat{a}_1$ | $\hat{a}_2$ |
| 30dB | 0.9613 | 0.4151 | -0.5585 |
| 20dB | 0.8057 | 0.4230 | -0.5645 |

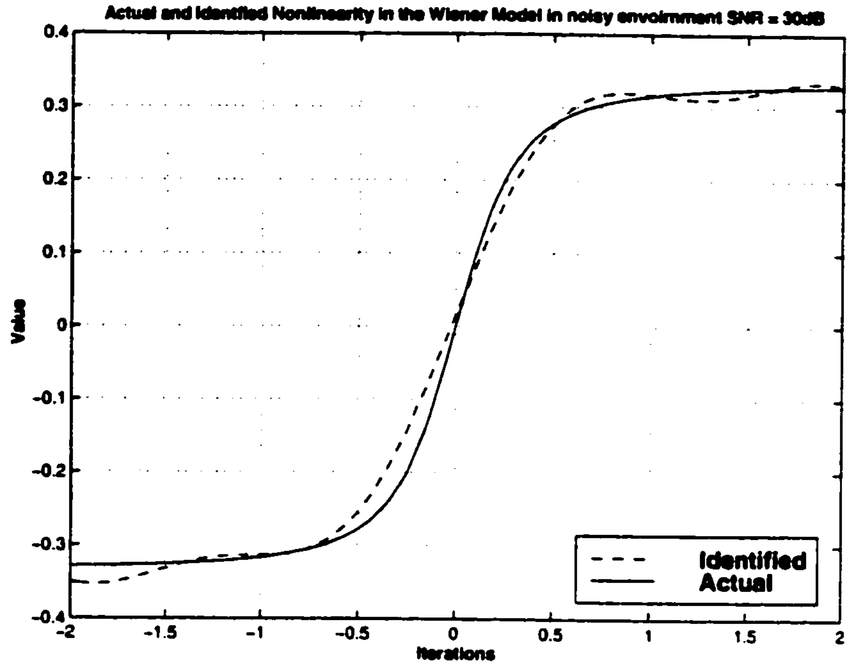


Figure 4.25: Actual and identified control valve in Wiener model with SNR 30dB.

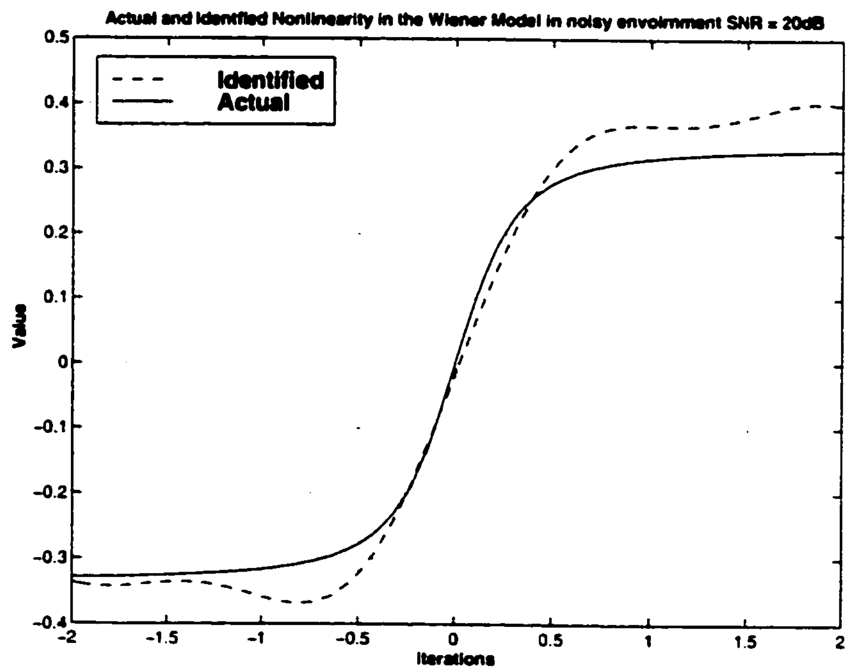


Figure 4.26: Actual and identified control valve in Wiener model with SNR 20dB.

# Chapter 5

# Development of Training Algorithms for MIMO Systems

MIMO nonlinear systems are very common in practice for example distillation column, chemical and biological processes. In the last two chapters training algorithms for SISO Hammerstein and Wiener systems were developed and validated by simulations. It was suggested in the beginning that the algorithm could be applied to MIMO systems as well. This chapter comprises of the development of training algorithms for MIMO Hammerstein and Wiener models which is the generalization of SISO cases.

# 5.1 MIMO Hammerstein System

A MIMO Hammerstein system can be classified into two classes with respect to the type of static nonlinearities in the system. The static nonlinearities can be separate or combined. In the following subsections training algorithm for both the cases are developed.

## 5.1.1 MIMO Hammerstein System with Separate Nonlinearities

Consider a general MIMO Hammerstein system with $M$-input, $M$-intermediate variables and $N$-output shown in Fig. 5.1. All the static nonlinearities are separate and each can be considered as an independent SISO nonlinear block. So, for $M$ nonlinearities at the input, there are $M$ inputs and $M$ intermediate variables.

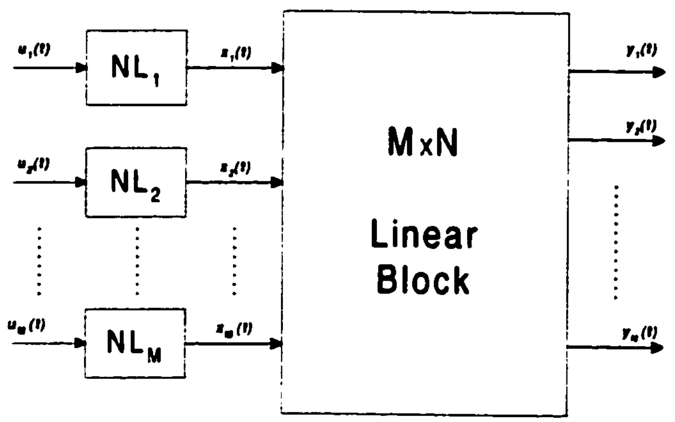


Figure 5.1: An $M$-input $N$-output Hammerstein system with separate nonlinearities.

The $M$ inputs given by $U(t) = [u_1(t)\ u_2(t)\ \dots\ u_M(t)]^T$ are fed to $M$ static nonlinearities modelled by RBFNN. The outputs of the static nonlinearities are:

$$X(t) = [x_1(t)\ x_2(t)\ \dots\ x_M(t)]^T, \tag{5.1}$$

and $x_i(t)$ estimated by the RBFNN is given by,

$$\hat{x}_i(t) = W_i\phi(\|u_i(t) - C_i\|). \tag{5.2}$$

The system output can be defined by:

$$Y(t) = [y_1(t)\ y_2(t)\ \dots\ y_N(t)]^T, \tag{5.3}$$

which is

$$Y(t) = \begin{bmatrix} G_{11} & G_{12} & \dots & G_{1M} \\ G_{21} & G_{22} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ G_{N1} & \dots & \dots & G_{NM} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_M(t) \end{bmatrix}. \tag{5.4}$$

The estimated outputs can be written in a similar fashion shown in algebraic form as:

$$\begin{aligned}
\hat{y}_1(t) &= \hat{x}_1(t)\hat{G}_{11} + \hat{x}_2(t)\hat{G}_{12} + \ldots + \hat{x}_M(t)\hat{G}_{1M}, \\
\hat{y}_2(t) &= \hat{x}_1(t)\hat{G}_{21} + \hat{x}_2(t)\hat{G}_{22} + \ldots + \hat{x}_M(t)\hat{G}_{2M}, \\
&\vdots \\
\hat{y}_N(t) &= \hat{x}_1(t)\hat{G}_{N1} + \hat{x}_2(t)\hat{G}_{N2} + \ldots + \hat{x}_M(t)\hat{G}_{NM},
\end{aligned}$$

where the transfer function corresponding to the $j^{th}$ intermediate variable and $i^{th}$ output is,

$$\hat{G}_{ij} = \frac{B_{ij}^{w}(q^{-1})}{A(q^{-1})}. \tag{5.5}$$

The polynomials $A(q^{-1})$ and $B_{ij}^{w}(q^{-1})$ are defined as,

$$\begin{aligned}
A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_n q^{-n}, \\
B_{ij}^{w}(q^{-1}) &= b_{0ij} + b_{1ij} q^{-1} + \cdots + b_{wij} q^{-w},
\end{aligned}$$

where $w$ is the order of zeros of that particular transfer function.

Now defining the error, keeping in mind that there are more than one outputs.

so the error will be a vector instead of a single value at any instant.

$$\begin{aligned}
e_1(t) &= y_1(t) - \hat{y}_1(t), \\
e_2(t) &= y_2(t) - \hat{y}_2(t), \\
&\vdots \\
e_N(t) &= y_N(t) - \hat{y}_N(t), \\
E(t) &= [e_1(t)\ e_2(t)\ \ldots\ e_N(t)]^T, \\
E(t) &= [(y_1(t) - \hat{y}_1(t))\ \ (y_2(t) - \hat{y}_2(t))\ \ldots\ (y_N(t) - \hat{y}_N(t))]^T.
\end{aligned}$$

This error will be used to update the parameters by developing the update equations based on LMS principle. The performance index as:

$$I = \frac{1}{2} E(t)^T E(t). \tag{5.6}$$

Following the method adopted in SISO case, the coefficients are augmented in $\theta$. The partial derivatives *w.r.t.* the coefficients and weights are found as following:

$$\frac{\partial I}{\partial \theta} = \frac{1}{2} \frac{\partial}{\partial \theta} \left( E(t)^T E(t) \right). \tag{5.7}$$

now differentiating $w.r.t.$ $a_i$ only,

$$\begin{aligned}\frac{\partial I}{\partial a_i} &= E(t)^T \frac{\partial}{\partial a_i}\left[(y_1(t)-\hat{y}_1(t)) \;\; (y_2(t)-\hat{y}_2(t)) \;\ldots\; (y_N(t)-\hat{y}_N(t))\right]^T, \\ &= -E(t)^T\left[\hat{y}_1(t-i) \;\; \hat{y}_2(t-i) \;\ldots\; \hat{y}_N(t-i)\right]^T. \end{aligned} \tag{5.8}$$

Keeping the coefficient in $A = [a_1 \; a_2 \; \ldots \; a_n]$ and the outputs in $\hat{Y}(t)$ as $\hat{Y}(t) = [\hat{Y}_1(t) \;\; \hat{Y}_2(t) \;\ldots\; \hat{Y}_N(t)]$, where $\hat{Y}_i(t) = [\hat{y}_1(t-i) \; \hat{y}_2(t-i) \;\ldots\; \hat{y}_N(t-i)\;]^T$ is the $i^{th}$ delay in the output vector. Eq. 5.8 becomes,

$$\frac{\partial I}{\partial A} = -E(t)^T\hat{Y}(t)$$

therefore, the update equation using Eq. 3.2 will be

$$\boxed{A(K+1) = A(K) + \alpha E(t)^T\hat{Y}(t).} \tag{5.9}$$

Now differentiating Eq. 5.7 $w.r.t$ $b_{ijk}$ $i.e.$ the $b_i$ for $k^{th}$ intermediate variable and $j^{th}$ output.

$$\begin{aligned}\frac{\partial I}{\partial b_{ijk}} &= E^T \frac{\partial}{\partial b_{ijk}}\left[(y_1(t)-\hat{y}_1(t)) \;\; (y_2(t)-\hat{y}_2(t)) \;\ldots\; (y_N(t)-\hat{y}_N(t))\right]^T, \\ &= -E(t)^T \frac{\partial}{\partial b_{ijk}}\left[\hat{y}_1(t) \;\; \hat{y}_2(t) \;\ldots\; \hat{y}_N(t)\right]^T, \\ &= -e_j(t)\frac{\partial}{\partial b_{ijk}}\hat{y}_j(t). \end{aligned}$$

$$\begin{aligned}\frac{\partial I}{\partial b_{ijk}} &= -e_j(t)\frac{\partial}{\partial b_{ijk}}\frac{B^w_{jk}(q^{-1})}{A(q^{-1})}\hat{x}_k(t),\\ \frac{\partial I}{\partial b_{ijk}} &= -e_j(t)\hat{x}_k(t-i).\end{aligned} \tag{5.10}$$

Keeping the coefficient in $B^w_{jk} = [b_{0jk}\ b_{1jk}\ \dots\ b_{wjk}]$ and the regressions of intermediate variable $\hat{x}_k(t)$ in $\hat{X}^w_{jk}(t) = [\hat{x}_k(t)\ \hat{x}_k(t-1)\ \dots\ \hat{x}_k(t-w)\ ]$. Therefore, Eq. 5.10 in stacked form will be,

$$\frac{\partial I}{\partial B^w_{jk}} = -e_j(t)\hat{X}^w_{jk}(t), \tag{5.11}$$

and the update Eq. 3.2 will take the form.

$$\boxed{B^w_{jk}(K+1) = B^w_{jk}(K) + \alpha E(t)^T\hat{X}^w_{jk}(t).} \tag{5.12}$$

Now finding the partial derivative *w.r.t.* weights of RBFNN. Taking Eq. 5.6.

$$\begin{aligned}I &= \frac{1}{2}E(t)^T E(t),\\ \frac{\partial I}{\partial W} &= \frac{1}{2}\frac{\partial}{\partial W}E(t)^T E(t).\end{aligned}$$

Differentiating *w.r.t.* any $W_i$ corresponding to $i^{th}$ nonlinearity only.

$$\frac{\partial I}{\partial W_i} = \frac{1}{2}\frac{\partial E(t)^T E(t)}{\partial W_i},$$

$$\begin{aligned}
\frac{\partial I}{\partial W_i} &= E(t)^T \frac{\partial E(t)}{\partial W_i}, \\
&= E(t)^T \frac{\partial}{\partial W_i} [(y_1(t) - \hat{y}_1(t)) \; (y_2(t) - \hat{y}_2(t)) \; \ldots \; (y_N(t) - \hat{y}_N(t))]^T, \\
&= -E(t)^T \frac{\partial}{\partial W_i} [\hat{y}_1(t) \; \hat{y}_2(t) \; \ldots \; \hat{y}_N(t)]^T. \qquad (5.13)
\end{aligned}$$

Just considering any $\hat{y}_j(t)$,

$$\begin{aligned}
\frac{\partial \hat{y}_j(t)}{\partial W_i} &= \frac{\partial}{\partial W_i} \left[ \frac{B_{j1}^{w1}(q^{-1})\hat{x}_1(t)}{A(q^{-1})} + \frac{B_{j2}^{w2}(q^{-1})\hat{x}_2(t)}{A(q^{-1})} + \cdots + \frac{B_{jM}^{wM}(q^{-1})\hat{x}_M(t)}{A(q^{-1})} \right], \\
&= \frac{\partial}{\partial W_i} \left[ \frac{B_{j1}^{w1}(q^{-1})W_1\phi_1(t)}{A(q^{-1})} + \cdots + \frac{B_{jM}^{wM}(q^{-1})W_M\phi_M(t)}{A(q^{-1})} \right], \\
&= \frac{\partial}{\partial W_i} \left[ B_{j1}^{w1}(q^{-1})W_1\phi_1(t) + \cdots + B_{jM}^{wM}(q^{-1})W_M\phi_M(t) \right], \\
&= B_{ji}^{wi}(q^{-1})\phi_i(t).
\end{aligned}$$

Therefore, Eq. 5.13 becomes,

$$\begin{aligned}
\frac{\partial I}{\partial W_i} &= -E(t)^T \frac{\partial}{\partial W_1} [\hat{y}_1(t) \; \hat{y}_2(t) \; \ldots \; \hat{y}_N(t)]^T, \\
&= -E(t)^T [B_{1i}^{w1}(q^{-1})\phi_i(t) \; B_{2i}^{w2}(q^{-1})\phi_i(t) \; \ldots \; B_{Ni}^{wN}(q^{-1})\phi_i(t)]^T. \\
&= -E^T(t) [B_{1i}^{w1}(q^{-1}) \; B_{2i}^{w2}(q^{-1}) \; \ldots \; B_{Ni}^{wN}(q^{-1})]^T \phi_i(t), \\
\frac{\partial I}{\partial W_i} &= -\sum_{j=1}^{N} e_j(t) B_{ji}^{wj}(q^{-1})\phi_i(t). \qquad (5.14)
\end{aligned}$$

Finally the update equation from Eq. 3.3 will be,

$$\boxed{W_i(K+1) = W_i(K) + \alpha \sum_{j=1}^{N} e_j(t) B_{ji}^{wj}(q^{-1})\phi_i(t).} \qquad (5.15)$$

### 5.1.2 MIMO Hammerstein System with Combined Nonlinearities

Consider a MIMO Hammerstein system having a combined static nonlinearity at the input, *i.e.*, there is only one static nonlinearity having $M$ inputs and $P$ outputs. Therefore, a general system of this type may have $M$ inputs, $N$ outputs and $P$ intermediate variables and is shown in Fig 5.2.

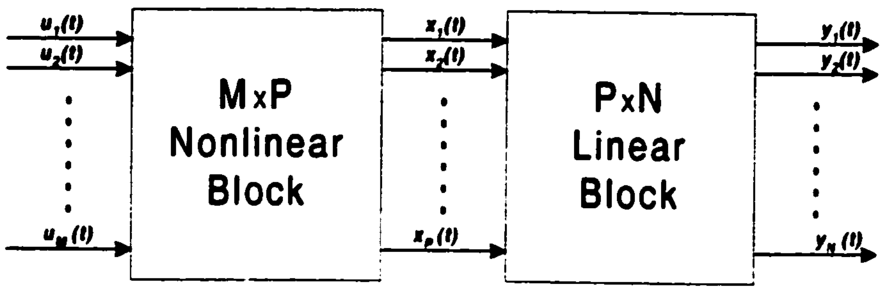


Figure 5.2: An $M$-input $N$-output Hammerstein system with combined nonlinearities.

The $M$ inputs given by, $U(t) = [u_1(t)\ u_2(t)\ \ldots\ u_M(t)]^T$ are fed to a static nonlinearity represented by RBFNN. The output of the static nonlinearity is given by,

$$X(t) = [x_1(t)\ x_2(t)\ \ldots\ x_M(t)]^T, \tag{5.16}$$

and $x_i(t)$ estimated by the RBFNN is defined by,

$$\hat{x}_i(t) = W_i\Phi(\|U(t) - C\|). \tag{5.17}$$

The vector $C$ comprises of the centers of the basis functions. It is to be noted that the effect of all the inputs is reflected over all the intermediate variables $x_i(t)$, as they depend on the input vector $U(t)$. Since, the combined static nonlinearity does not have any effect on the linear part, The system output can be represented by Eq. 5.3 and Eq. 5.4, where all the terms are already defined.

Now differentiating the performance index, given in Eq. 5.6, *w.r.t.* $a_i$,

$$\frac{\partial I}{\partial a_i} = E(t)^T \frac{\partial}{\partial a_i}[(y_1(t) - \hat{y}_1(t)) \ (y_2(t) - \hat{y}_2(t)) \ \ldots \ (y_N(t) - \hat{y}_N(t))]^T,$$

The linear part in a Hammerstein model with separate or combined static nonlinearities remains the same. and is not effected by the type of the nonlinearity. being at the output. Therefore. Eq. 5.9 is also true for the combined static nonlinearity case. and updates the parameters $a_i$ as.

$$\boxed{A(K+1) = A(K) + \alpha E(t)^T \hat{Y}(t).} \tag{5.18}$$

Similarly the coefficients $b_{ijk}$ stacked in $B^w_{jk}$ are also updated using Eq. 5.12. *i.e.*

$$\boxed{B^w_{jk}(K+1) = B^w_{jk}(K) + \alpha E(t)^T \hat{Y}(t).} \tag{5.19}$$

Now finding the update equation for weights of RBFNN. Taking Eq. 5.6.

$$
\begin{aligned}
I &= \frac{1}{2}E(t)^T E(t), \\
\frac{\partial I}{\partial W} &= \frac{1}{2}\frac{\partial}{\partial W}E(t)^T E(t).
\end{aligned}
$$

Differentiating *w.r.t.* any $W_i$ corresponding to $i^{th}$ intermediate variable $x_i(t)$ only,

$$
\begin{aligned}
\frac{\partial I}{\partial W_i} &= \frac{1}{2}\frac{\partial E(t)^T E(t)}{\partial W_i}, \\
&= E(t)^T \frac{\partial E(t)}{\partial W_i}, \\
&= E(t)^T \frac{\partial}{\partial W_i}[(y_1(t) - \hat{y}_1(t))\ (y_2(t) - \hat{y}_2(t))\ \ldots\ (y_N(t) - \hat{y}_N(t))]^T, \\
&= -E(t)^T \frac{\partial}{\partial W_i}[\hat{y}_1(t)\ \hat{y}_2(t)\ \ldots\ \hat{y}_N(t)]^T, \qquad (5.20)
\end{aligned}
$$

just differentiating any $\hat{y}_j(t)$ *w.r.t.* $W_i$

$$
\begin{aligned}
\frac{\partial \hat{y}_j(t)}{\partial W_i} &= \frac{\partial}{\partial W_i}\left[\frac{B_{j1}^{w1}(q^{-1})\hat{x}_1(t)}{A(q^{-1})} + \frac{B_{j2}^{w2}(q^{-1})\hat{x}_2(t)}{A(q^{-1})} + \cdots + \frac{B_{jM}^{wM}(q^{-1})\hat{x}_M(t)}{A(q^{-1})}\right], \\
&= \frac{\partial}{\partial W_i}\left[\frac{B_{j1}^{w1}(q^{-1})W_1\Phi(t)}{A(q^{-1})} + \cdots + \frac{B_{jM}^{wM}(q^{-1})W_M\phi(t)}{A(q^{-1})}\right]. \\
&= \frac{\partial}{\partial W_i}\left[B_{j1}^{w1}(q^{-1})W_1\Phi(t) + \cdots + B_{jM}^{wM}(q^{-1})W_M\Phi(t)\right]. \\
\frac{\partial \hat{y}_j(t)}{\partial W_i} &= B_{ji}^{wi}(q^{-1})\Phi(t). \qquad (5.21)
\end{aligned}
$$

Now Eq. 5.20 using Eq. 5.21 becomes,

$$\begin{aligned}
\frac{\partial I}{\partial W_i} &= -E(t)^T \frac{\partial}{\partial W_1}[\hat{y}_1(t)\ \hat{y}_2(t)\ \ldots\ \hat{y}_N(t)]^T, \\
&= -E(t)^T[B_{1i}^{w1}(q^{-1})\Phi(t)\ B_{2i}^{w2}(q^{-1})\Phi(t)\ \ldots\ B_{Ni}^{wN}(q^{-1})\Phi(t)]^T, \\
&= -E^T(t)[B_{1i}^{w1}(q^{-1})\ B_{2i}^{w2}(q^{-1})\ \ldots\ B_{Ni}^{wN}(q^{-1})]^T\Phi(t), \\
\frac{\partial I}{\partial W_i} &= -\sum_{j=1}^{N} e_j(t)B_{ji}^{wj}(q^{-1})\Phi(t).
\end{aligned} \tag{5.22}$$

Finally the weight update equation from Eq. 3.3 will be,

$$\boxed{W_i(K+1) = W_i(K) + \alpha\sum_{j=1}^{N} e_j(t)B_{ji}^{wj}(q^{-1})\Phi(t).} \tag{5.23}$$

## 5.2 MIMO Wiener Systems

In this section training algorithm for MIMO Wiener systems will be developed. Similar to Hammerstein systems, Wiener systems are also classified according to the type of static nonlinearity *i.e.* separate and combined. Following are the development of training algorithm and update equations for MIMO Wiener systems.

### 5.2.1 MIMO Wiener System with Separate Nonlinearities

Consider $M$-input $N$-output system shown in Fig. 5.3. There are $N$ separate static nonlinearities at the output, fed by the same number of intermediate variables. The $N$ intermediate variables are obtained by feeding the $M$ inputs to the linear dynamic block.

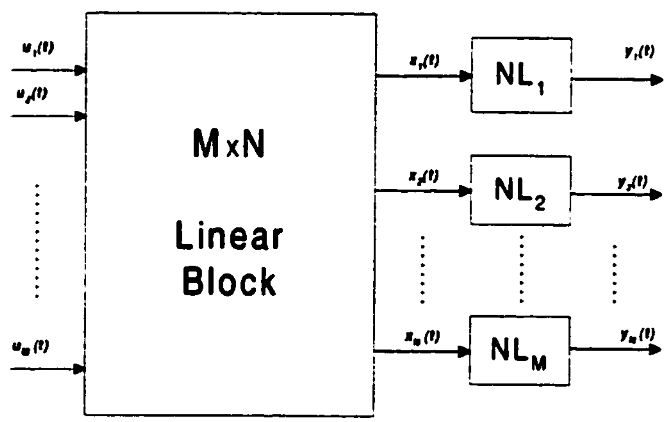


Figure 5.3: An $M$-input $N$-output Wiener system with separate nonlinearities.

The number of intermediate variables $x(t)$ is the same as the number of outputs of the system, since, there are as much nonlinearities as the number of intermediate variables.

$$X(t) = [x_1(t)\ x_2(t)\ \dots\ x_N(t)]^T, \tag{5.24}$$

defined by,

$$X(t) = \begin{bmatrix} G_{11} & G_{12} & \dots & G_{1M} \\ G_{21} & G_{22} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ G_{N1} & \dots & \dots & G_{NM} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_M(t) \end{bmatrix} \tag{5.25}$$

The system outputs are given by $Y(t) = [y_1(t)\ y_2(t)\ \dots\ y_N(t)]$, where the estimate of $y_j(t)$ is given by,

$$\hat{y}_j(t) = W_j \phi(\|\hat{X}(t) - C_i\|).$$

The partial derivatives of the performance index given by Eq. 5.7 *w.r.t.* the weights and coefficients are found as follows.

$$\begin{aligned} \frac{\partial I}{\partial W} &= \frac{1}{2}\frac{\partial}{\partial W}(E(t)^T E(t)), \qquad (5.26) \\ &= E^T(t)\frac{\partial}{\partial W}[(y_1(t) - \hat{y}_1(t))\ (y_2(t) - \hat{y}_2(t))\ \dots\ (y_N(t) - \hat{y}_N(t))], \\ &= -E^T(t)\frac{\partial}{\partial W}[\hat{y}_1(t)\ \hat{y}_2(t)\ \dots\ \hat{y}_N(t)]. \qquad (5.27) \end{aligned}$$

considering the derivative $w.r.t. W_i$ only,

$$\begin{aligned}
\frac{\partial I}{\partial W_i} &= -E^T(t)\frac{\partial}{\partial W_i}[\hat{y}_1(t)\ \hat{y}_2(t)\ \ldots\ \hat{y}_N(t)], \\
&= -E^T(t)\frac{\partial}{\partial W_i}[W_1\phi_1(t)\ W_2\phi_2(t)\ \ldots\ W_i\phi_N(t)], \\
&= -e_i(t)\phi_i(t).
\end{aligned}$$

Defining $\Phi(t) = [\phi_1(t)\ \phi_2(t)\ \ldots\ \phi_N(t)]^T$. Therefore, 5.27 becomes,

$$\frac{\partial I}{\partial W} = -E^T(t)\Phi(t) \tag{5.28}$$

Eq. 5.28, together with Eq. 3.3 gives the final weight update equation.

$$\boxed{W(K+1) = W(K) + \alpha E^T(t)\Phi(t)} \tag{5.29}$$

Now finding the derivatives with respect to the coefficients.

$$\begin{aligned}
\frac{\partial I}{\partial \theta} &= \frac{1}{2}\frac{\partial}{\partial \theta}(E(t)^T E(t)), \\
&= E^T(t)\frac{\partial}{\partial \theta}[(y_1(t) - \hat{y}_1(t))\ (y_2(t) - \hat{y}_2(t))\ \ldots\ (y_N(t) - \hat{y}_N(t))]^T, \\
&= -E^T(t)\frac{\partial}{\partial \theta}[\hat{y}_1(t)\ \hat{y}_2(t)\ \ldots\ \hat{y}_N(t)]^T. \qquad (5.30)
\end{aligned}$$

Considering only the partial derivative $w.r.t.\ a_i$,

$$\frac{\partial I}{\partial a_i} = -E^T(t)\frac{\partial}{\partial a_i}\left[\hat{y}_1(t)\ \hat{y}_2(t)\ \ldots\ \hat{y}_N(t)\right]^T, \tag{5.31}$$

and now differentiating only $\hat{y}_j$,

$$\frac{\partial \hat{y}_j}{\partial a_i} = \frac{\partial W_j \phi_j(t)}{\partial a_i}.$$

The above equation can be taken as the derivative for single output. The derivative for single output is derived in chapter 3. For the MIMO system for the $j^{th}$ intermediate variable and $j^{th}$ output, the derivative $w.r.t.\ a_i$ will be:

$$\frac{\partial \hat{y}_j}{\partial a_i} = -\frac{2}{\beta_j^2}\hat{x}_j(t-i)\sum_{l=1}^{n_j}(\hat{x}_j(t) - c_{jl})w_{jl}\phi(\|\hat{x}_j(t) - c_{jl}\|). \tag{5.32}$$

Stacking from Eq. 5.32 in Eq. 5.31 to obtain.

$$\frac{\partial I}{\partial a_i} = -E^T(t)\begin{bmatrix} -\frac{2}{\beta_1^2}\hat{x}_1(t-1)\sum_{l=1}^{n_1}(\hat{x}_1(t) - c_{1l})w_{1l}\phi(\|\hat{x}_1(t) - c_{1l}\|) \\ \vdots \\ -\frac{2}{\beta_j^2}\hat{x}_j(t-i)\sum_{l=1}^{n_j}(\hat{x}_j(t) - c_{jl})w_{jl}\phi(\|\hat{x}_j(t) - c_{jl}\|) \\ \vdots \\ -\frac{2}{\beta_N^2}\hat{x}_N(t-1)\sum_{l=1}^{n_N}(\hat{x}_N(t) - c_{Nl})w_{Nl}\phi(\|\hat{x}_N(t) - c_{Nl}\|) \end{bmatrix} \tag{5.33}$$

$$\frac{\partial I}{\partial a_i} = 2\sum_{j=1}^{N}\left(\frac{e_j(t)}{\beta_j^2}\hat{x}_j(t-i)\sum_{l=1}^{n_j}(\hat{x}_j(t) - c_{jl})w_{jl}\phi(\|\hat{x}_j(t) - c_{jl}\|)\right), \tag{5.34}$$

and the update equation for $a_i$ using Eq. 5.34 and Eq. 3.2 will be,

$$\boxed{a_i(K+1) = a_i(K) - 2\alpha\sum_{j=1}^{N}\left(\frac{e_j(t)}{\beta_j^2}\hat{x}_j(t-i)\sum_{l=1}^{n_j}(\hat{x}_j(t) - c_{jl})w_{jl}\phi_j(\|\hat{x}_j(t) - c_{jl}\|)\right).} \tag{5.35}$$

Similarly from the SISO case the, derivative *w.r.t.* $b_{ijk}$ *i.e.* $b_i$ for $k_{th}$ input and $j_{th}$ intermediate variable will be,

$$\frac{\partial I}{\partial b_{ijk}} = \frac{2e_j(t)}{\beta_j^2}u_k(t-i)\sum_{l=1}^{n_j}(\hat{x}_j(t) - c_{jl})w_{jl}\phi(\|\hat{x}_j(t) - c_{jl}\|), \tag{5.36}$$

and the update equation using Eq. 3.2 and Eq. 5.36.

$$\boxed{b_{ijk}(K+1) = b_{ijk}(K) - \frac{2\alpha e_j(t)}{\beta_j^2}u_k(t-i)\sum_{l=1}^{n_j}(\hat{x}_j(t) - c_{jl})w_{jl}\phi(\|\hat{x}_j(t) - c_{jl}\|).} \tag{5.37}$$

### 5.2.2 MIMO Wiener System with Combined Nonlinearties

Consider an $M$-input $N$-output Wiener system with a combined nonlinearity at the output show in Fig. 5.4. This means, that there is only one nonlinearity with $N$ outputs fed by, say, $P$ intermediate variables, that are the output of the $M$ input linear block.

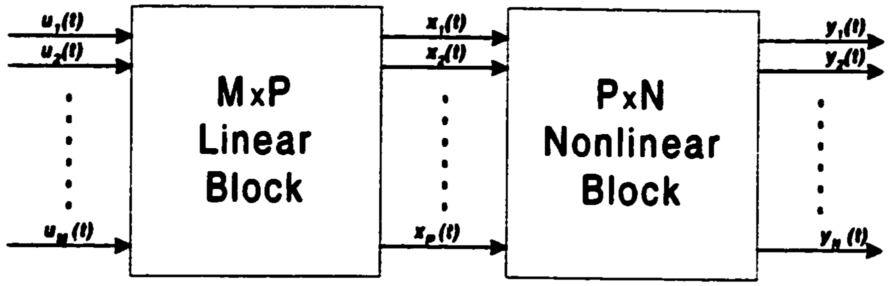


Figure 5.4: An $M$-input $N$-output Wiener system with combined nonlinearities.

The intermediate variables are,

$$X(t) = [x_1(t)\ x_2(t)\ \dots\ x_P(t)]^T. \tag{5.38}$$

defined by,

$$X(t) = \begin{bmatrix} G_{11} & G_{12} & \dots & G_{1M} \\ G_{21} & G_{22} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ G_{P1} & \dots & \dots & G_{PM} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_M(t) \end{bmatrix} \tag{5.39}$$

The system outputs are given by $\hat{Y}(t) = [\hat{y}_2(t)\ \hat{y}_N(t)\ \dots\ \hat{y}_N(t)]$. For a combined nonlinearity case there is only be one $P$ input $N$ output nonlinearity whose output is estimated by,

$$\hat{y}_j(t) = W_j \Phi(\|\hat{X}(t) - C\|),$$

and $C = [c_1\ c_2\ \dots\ c_{n_o}]$ where $n$ is the number of centers in the RBFNN. Keeping the performance index as defined in Eq. 5.6.

$$I = \frac{1}{2}E(t)^T E(t). \tag{5.40}$$

The partial derivatives *w.r.t.* the weights and coefficients are found as following.

$$\begin{aligned}
\frac{\partial I}{\partial W} &= \frac{1}{2}\frac{\partial}{\partial W}(E(t)^T E(t)), \\
&= E^T(t)\frac{\partial}{\partial W}\left[(y_1(t) - \hat{y}_1(t))\ (y_2(t) - \hat{y}_2(t))\ \ldots\ (y_N(t) - \hat{y}_N(t))\right], \\
&= -E^T(t)\frac{\partial}{\partial W}\left[\hat{y}_1(t)\ \hat{y}_2(t)\ \ldots\ \hat{y}_N(t)\right].
\end{aligned} \tag{5.41}$$

Considering only the derivative *w.r.t.* $W_j$,

$$\begin{aligned}
\frac{\partial I}{\partial W_j} &= -E^T(t)\frac{\partial}{\partial W_j}\left[\hat{y}_1(t)\ \hat{y}_2(t)\ \ldots\ \hat{y}_N(t)\right], \\
&= -\left(-e_1(t)\frac{\partial}{\partial W_j}\hat{y}_1(t)\ - e_2(t)\frac{\partial}{\partial W_j}\hat{y}_2(t)\ \ldots\ - e_N(t)\frac{\partial}{\partial W_j}\hat{y}_N(t)\right), \\
&= -\left(-e_1(t)\frac{\partial}{\partial W_j}W_1\Phi(t)\ - e_2(t)\frac{\partial}{\partial W_j}W_2\Phi(t)\ \ldots\ - e_N(t)\frac{\partial}{\partial W_j}W_N\Phi(t)\right). \\
\frac{\partial I}{\partial W_j} &= -e_j(t)\Phi(t).
\end{aligned} \tag{5.42}$$

With the result in Eq. 5.42, Eq. 5.41 becomes,

$$\frac{\partial I}{\partial W} = -E^T(t)\Phi(t). \tag{5.43}$$

and the update Eq. 3.3 finally becomes,

$$\boxed{W(K+1) = W(K) + \alpha E^T(t)\Phi(t).} \tag{5.44}$$

Now developing the update equations for the coefficients of the linear part. The performance index defined in Eq. 5.7 is differentiated *w.r.t.* $\theta$.

$$\begin{aligned}
\frac{\partial I}{\partial \theta} &= \frac{1}{2}\frac{\partial}{\partial \theta}(E(t)^T E(t)), \\
&= E(t)^T \frac{\partial E(t)}{\partial \theta}, \\
\frac{\partial I}{\partial \theta} &= -E^T(t)\frac{\partial}{\partial \theta}\left[\hat{y}_1(t)\ \hat{y}_2(t)\ \ldots\ \hat{y}_N(t)\right]^T,
\end{aligned} \tag{5.45}$$

and now differentiating only $\hat{y}_j$.

$$\begin{aligned}
\frac{\partial \hat{y}_j}{\partial \theta} &= \frac{\partial W_j \Phi(t)}{\partial \theta}, \\
&= \frac{\partial W_j \Phi(\|\hat{X}(t) - C\|)}{\partial \theta}, \\
&= \frac{\partial}{\partial \theta} W_j \exp\left(-\frac{\|\hat{X}(t) - C\|^2}{\beta^2}\right), \\
&= W_j \exp\left(-\frac{\|\hat{X}(t) - C\|^2}{\beta^2}\right) \frac{\partial}{\partial \theta}\left(-\frac{\|\hat{X}(t) - C\|^2}{\beta^2}\right),
\end{aligned}$$

$$
\begin{aligned}
&= \frac{-W_j\Phi(t)}{\beta^2}\frac{\partial}{\partial\theta}\left(\sqrt{((\hat{X}(t)-c_1)^2\ +\ \ldots\ +\ (\hat{X}(t)-c_{n_o})^2)}^{\,2}\right), \\
&= \frac{-W_j\Phi(t)}{\beta^2}\left(2(\hat{X}(t)-c_1)\frac{\partial}{\partial\theta}\hat{X}(t)\ +\ \ldots\ +\ 2(\hat{X}(t)-c_{n_o})\frac{\partial}{\partial\theta}\hat{X}(t)\right).
\end{aligned}
\tag{5.46}
$$

Just considering $\frac{\partial}{\partial\theta}X(t)$,

$$
\frac{\partial}{\partial\theta}\hat{X}(t) = \frac{\partial}{\partial\theta}[\hat{x}_1(t)\ \hat{x}_2(t)\ \ldots\ \hat{x}_P(t)]^T. \tag{5.47}
$$

Now just taking the $\frac{\partial}{\partial\theta}\hat{x}_k(t)$ and differentiating *w.r.t.* $a_i$,

$$
\begin{aligned}
\frac{\partial}{\partial a_i}\hat{x}_k(t) &= \frac{\partial}{\partial a_i}\left(A(q^{-1})\hat{x}_k(t) + B_{k1}^{w1}(q^{-1})u_1(t) +\ \ldots\ + B_{kM}^{wM}(q^{-1})u_M(t)\right) \\
\frac{\partial}{\partial a_i}\hat{x}_k(t) &= \hat{x}_k(t-i).
\end{aligned}
\tag{5.48}
$$

Stacking $\frac{\partial}{\partial a_i}\hat{x}_k(t)$ in $\frac{\partial}{\partial a_i}\hat{X}(t)$.

$$
\frac{\partial}{\partial a_i}\hat{X}(t) = [\hat{x}_1(t-i)\ \hat{x}_2(t-i)\ \ldots\ \hat{x}_P(t-i)]^T. \tag{5.49}
$$

Defining $\nabla\hat{X}(t-i)$ as $\left[\frac{\partial}{\partial a_i}\hat{X}(t)\right]^T$ and then putting in Eq. 5.46 for $a_i$,

$$
\begin{aligned}
\frac{\partial\hat{y}_j}{\partial a_i} &= \frac{-W_j\Phi(t)}{\beta^2}\left(2(\hat{X}(t)-c_1)\nabla(\hat{X}(t-i))\ +\ \ldots\ +\ 2(\hat{X}(t)-c_{n_o})\nabla\hat{X}(t-i)\right). \\
&= \frac{-2W_j\Phi(t)}{\beta^2}\left((\hat{X}(t)-c_1)\ +\ \ldots\ +\ (\hat{X}(t)-c_{n_o})\right)\nabla\hat{X}(t-i). \\
\frac{\partial\hat{y}_j}{\partial a_i} &= \frac{-2W_j\Phi(t)}{\beta^2}\left(\sum_{n=1}^{n_o}\hat{X}(t)-c_n\right)\nabla\hat{X}(t-i).
\end{aligned}
\tag{5.50}
$$

Now, with $W = [W_1 \; W_2 \; \ldots \; W_N]$, the expression given by the Eq. 5.50 is stacked in Eq. 5.45,

$$\begin{aligned}\frac{\partial I}{\partial a_i} &= -E^T(t)\left(\frac{-2W_j\Phi(t)}{\beta^2}\left(\sum_{n=1}^{n_o}\hat{X}(t)-c_n\right)\nabla\hat{X}(t-i)\right),\\ &= \frac{2}{\beta^2}E^T(t)W_j\Phi(t)\left(\sum_{n=1}^{n_o}\hat{X}(t)-c_n\right)\nabla\hat{X}(t-i).\end{aligned} \tag{5.51}$$

This equation can also be written by expanding the vectors $[\hat{X}(t) - c_n]$ and $\nabla\hat{X}(t-i)$,

$$\frac{\partial I}{\partial a_i} = \frac{2}{\beta^2}E^T(t)W\Phi(t)\left(\sum_{n=1}^{n_o}\sum_{k=1}^{P}(\hat{x}_k(t)-c_n)(\hat{x}_k(t-i))\right), \tag{5.52}$$

and the final update equation becomes,

$$\boxed{a_i(K+1) = a_i(K) - \frac{2}{\beta^2}\alpha E^T(t)W\Phi(t)\left(\sum_{n=1}^{n_o}\sum_{k=1}^{P}(\hat{x}_k(t)-c_n)(\hat{x}_k(t-i))\right).} \tag{5.53}$$

The partial derivatives *w.r.t.* $b_{ijk}$ can be found in similar fashion proceeding from Eq. 5.46 and differentiating *w.r.t.* $b_{ijk}$. It is evident from Eq. 5.48 that,

$$\frac{\partial}{\partial b_{ijk}}\hat{x}_k(t) = u_j(t-i). \tag{5.54}$$

Since the coefficient $b_{ijk}$ is only for the $k^{th}$ input and $j^{th}$ intermediate variable

$\hat{x}_k(t)$, therefore Eq. 5.47 for $b_{ijk}$ contains only one term given in Eq. 5.54.

$$\frac{\partial}{\partial b_{ijk}}\hat{X}(t) = u_j(t-i) \tag{5.55}$$

Now Eq. 5.46 for $b_{ijk}$ becomes

$$\begin{aligned} \frac{\partial \hat{y}_l}{\partial b_{ijk}} &= \frac{-W_l\Phi(t)}{\beta^2}\left(2(\hat{X}(t)-c_1)u_j(t-i) + \ldots + 2(\hat{X}(t)-c_{n_o})u_j(t-i)\right), \\ &= \frac{-W_l\Phi(t)}{\beta^2}\left(2(\hat{X}(t)-c_1) + \ldots + 2(\hat{X}(t)-c_{n_o})\right)u_j(t-i), \\ &= \frac{-2W_l\Phi(t)}{\beta^2}\left(\sum_{n=1}^{n_o}\hat{X}(t)-c_n\right)u_j(t-i). \end{aligned} \tag{5.56}$$

The partial derivative of the performance index $w.r.t.$ $b_{ijk}$ will take the form

$$\frac{\partial I}{\partial b_{ijk}} = 2E^T(t)\begin{bmatrix} \frac{-W_1\Phi(t)}{\beta^2}\sum_{n=1}^{n_o}(\hat{X}(t)-c_n)u_j(t-i) \\ \vdots \\ \frac{-W_l\Phi(t)}{\beta^2}\sum_{n=1}^{n_o}(\hat{X}(t)-c_n)u_j(t-i) \\ \vdots \\ \frac{-W_N\Phi(t)}{\beta^2}\sum_{n=1}^{n_o}(\hat{X}(t)-c_n)u_j(t-i) \end{bmatrix} \tag{5.57}$$

Since $W = [W_1\ W_2\ \ldots\ W_N]^T$, the above expression can be written as,

$$\begin{aligned}\frac{\partial I}{\partial b_{ijk}} &= \frac{2}{\beta^2} E^T(t) W \Phi(t) \sum_{n=1}^{n_o} (\hat{X}(t) - c_n) u_j(t-i), \\ &= \frac{2u_j(t-i)}{\beta^2} E^T(t) W \Phi(t) \sum_{n=1}^{n_o} (\hat{X}(t) - c_n). \end{aligned} \tag{5.58}$$

Finally Eq. 5.58 combined with the LMS parameter update Eq. 3.2 becomes,

$$\boxed{b_{ijk}(K+1) = b_{ijk}(K) - \frac{2\alpha\ u_j(t-i)}{\beta^2} E^T(t) W \Phi(t) \sum_{n=1}^{n_o} (\hat{X}(t) - c_n).} \tag{5.59}$$

# Chapter 6

# Simulation Results for MIMO systems

In this chapter the results obtained for the MIMO Hammerstein and Wiener systems are verified using simulation examples. The examples considered are 2-input 2-output systems. However, the algorithms are valid for any number of inputs and outputs. Practical examples of heat exchanger and control valve with the saturation nonlinearity are used for the simulations.

## 6.1 Simulation Results for MIMO Hammerstein systems

A 2-input 2-output example is considered here to verify the training algorithm for MIMO Hammerstein systems. One of the static nonlinearities is of a heat exchanger model and the other is a saturation nonlinearity. The characteristics of the saturation nonlinearity are defined in Eq. 4.2 as,

$$x_1(t) = \begin{cases} 0.5, & \text{for } u_1(t) > 0.5 \\ u_1(t), & \text{for } -0.5 \leq u_1(t) \leq 0.5 \\ -0.5, & \text{for } u_1(t) < -0.5 \end{cases} \tag{6.1}$$

The heat exchanger is defined in Eq. 4.8 as,

$$x_2(t) = -31.549u_2(t) + 41.732u_2^2(t) - 24.201u_2^3(t) + 68.634u_2^4(t) \tag{6.2}$$

As discussed while developing the update equation for the MIMO systems. the order of the poles is the same for all the transfer function in the linear block. this means all the outputs have the same number of delayed regressions. In this example second order linear systems are considered, given by,

$$y_1(t) = 0.62y_1(t-1) - 0.75y_1(t-2) + 0.81x_1(t) - 0.53x_2(t), \quad (6.3)$$

$$y_2(t) = 0.62y_2(t-1) - 0.75y_2(t-2) + 0.62x_1(t) - 0.22x_2(t). \quad (6.4)$$

Using random numbers uniformly generated in the interval $[-2, 2]$, the desired outputs are produced by using the process model. The proposed identification scheme comprised of two RBFNNs in series with a MIMO ARMA model. The centers for both the RBFNNs are evenly located in the input space. The width of the basis function for the saturation nonlinearity is kept as 0.5 and for the heat exchanger as 0.6. The learning rate was 0.04. These values are selected after few trial runs and observing the SISO examples. The linear part was modeled by the following MIMO ARMA model,

$$y_1(t) = a_1y_1(t-1) - a_2y_1(t-2) + b_{011}x_1(t)b_{012}x_2(t),$$

$$y_2(t) = a_1y_2(t-1) - a_2y_2(t-2) + b_{021}x_1(t)b_{022}x_2(t).$$

The proposed algorithm is applied to update the parameters of the identification scheme. The RBFNNs identified the static nonlinearities very accurately. The actual and identified nonlinearities are shown in Fig. 6.1 and Fig. 6.2. The ARMA models estimated the linear dynamics of the system. The values of the parameters $a_1$ and

$a_2$ were converged to 0.6199 and -0.7512, respectively, which are very close to the actual ones. The true values for the zeros $b_{011}$, $b_{012}$, $b_{021}$ and $b_{022}$ for each of the transfer function were 0.81, 0.62, -0.53 and -0.22 that converged to 0.5333, 0.5306, -0.3568 and -0.1842, respectively. These values differ from the true ones, as noted in the SISO examples of Hammerstein model. The reason for this variation is described in example 1 of Hammerstein model. The mean square error is minimized to 0.25 after around 80 iterations. The mean square error plot is shown in Fig. 6.4. The identified static nonlinearities and truly identified poles of the linear parts reveal the capability of the identification algorithm to be applied on MIMO systems.

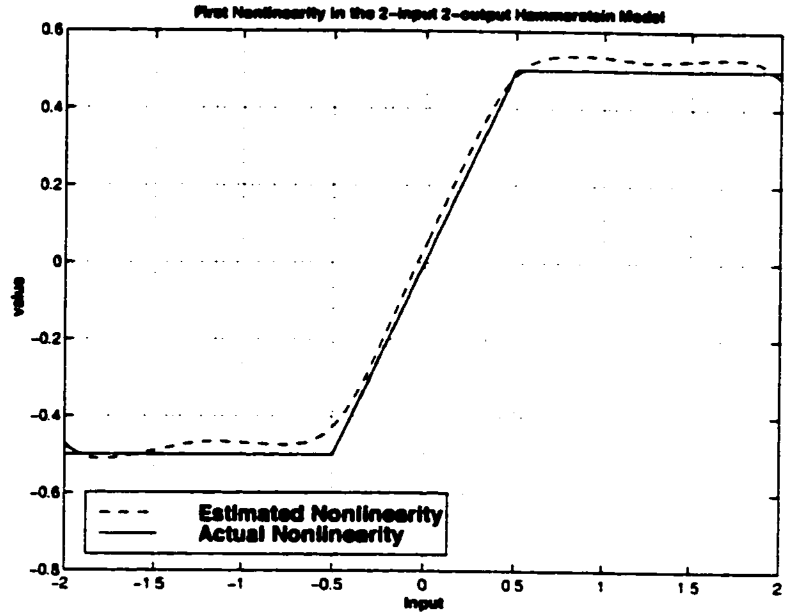


Figure 6.1: First nonlinearity in the 2-input 2-output Hammerstein system

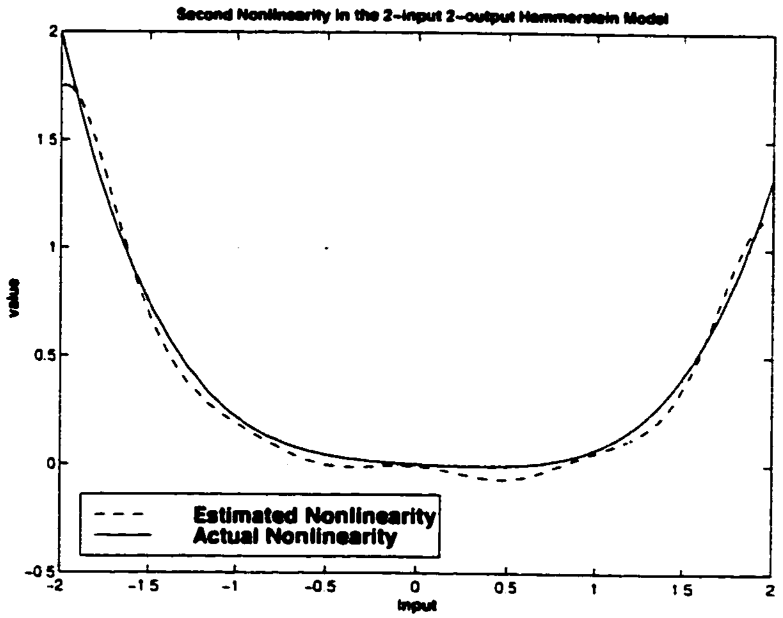


Figure 6.2: Second nonlinearity in the 2-input 2-output Hammerstein system

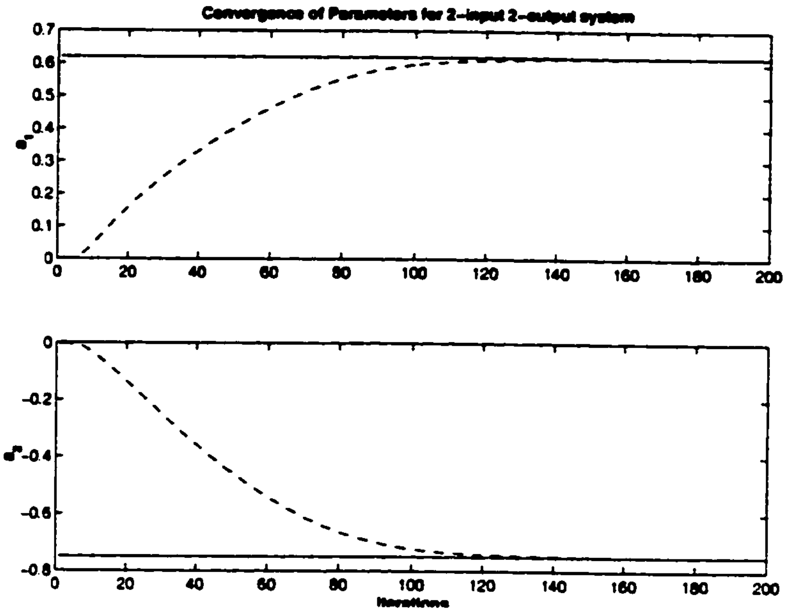

Figure 6.3: Convergence of poles of the linear part in 2-input 2output Hammerstein system

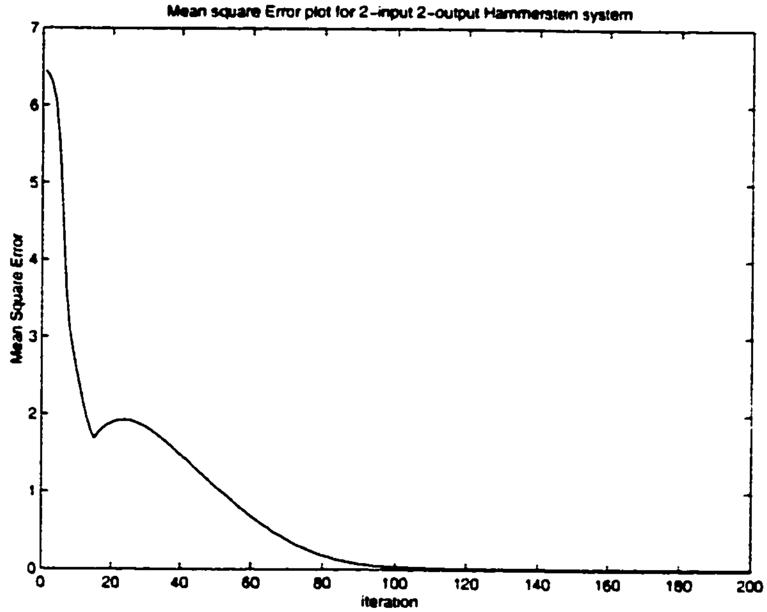


Figure 6.4: Square error plot for the 2-input 2-output Hammerstein system

## 6.2 Simulation results for MIMO Wiener Systems

To validate the update equations for MIMO Wiener system, a 2-input 2-output example is considered. The static nonlinearities chosen are a saturation nonlinearity and a control valve. The characteristics of the saturation nonlinearity are given by,

$$y_1(t) = \left\{ \begin{array}{ll} 0.5, & \text{for } x_1(t) > 0.5 \\ x_1(t), & \text{for } -0.5 \le x_1(t) \le 0.5 \\ -0.5, & \text{for } x_1(t) < -0.5 \end{array} \right\}$$

and the control valve is defined as,

$$y_2(t) = \frac{x_1(t)}{\sqrt{0.10 + 0.90{x_1}^2(t)}}. \tag{6.5}$$

As discussed while developing the update equation for the MIMO systems. the poles are the same for all the transfer functions in the linear block. this means all the outputs have the same number of delayed regressions. A third order linear systems is considered in this example as,

$$\begin{aligned} x_1(t) &= 0.35x_1(t-1) - 0.65x_1(t-2) + 0.15x_1(t-3) + 1.81u_1(t) - 0.85u_2(t). \\ x_2(t) &= 0.35x_2(t-1) - 0.65x_2(t-2) + 0.15x_2(t-3) + 1.62u_1(t) - 1.22u_2(t). \end{aligned}$$

The identification scheme comprised of a MIMO ARMA model and two RBFNNs at the output. The ARMA model is given by,

$$
\begin{aligned}
x_1(t) &= a_1x_1(t-1) + a_2x_1(t-2) + a_3x_1(t-3) + b_{011}u_1(t)b_{012}u_2(t), \\
x_2(t) &= a_1x_2(t-1) + a_2x_2(t-2) + a_3x_2(t-3) + b_{021}u_1(t)b_{022}u_2(t).
\end{aligned}
$$

The centers of the RBFNN modeling the saturation nonlinearity are located evenly in the range $[-2.5, 2.5]$ and the width is kept as 0.75. The RBFNN modeling the control valve is centered at $[-3, -2, -1, -0.08, 0.08, 1, 2, 3]$. The width is kept as 0.95. These values are selected after few trial runs.

The proposed identification algorithm is applied to update the coefficients of ARMA and the weights of the RBFNNs. The values of the coefficients $a_1$, $a_2$ and $a_3$ converged to 0.3483, -0.6842 and -0.1612, respectively. The true values for the zeros $b_{011}$, $b_{012}$, $b_{021}$ and $b_{022}$ for each of the transfer function are 1.81, 1.62, -0.85 and -1.22 that converged to 0.5333, 0.5306, -0.3568 and -0.1842, respectively. These values differ from the true ones, since these zeros can simply be considered as gains and actually are compensated with the nonlinearities as discussed in example 1 of the Wiener model. The actual and identified nonlinearities are shown in Fig. 6.5 and Fig. 6.6, reflecting the nice identification capability of the proposed scheme for MIMO systems. This example shows the generalization capability of the training algorithm for MIMO Wiener systems.

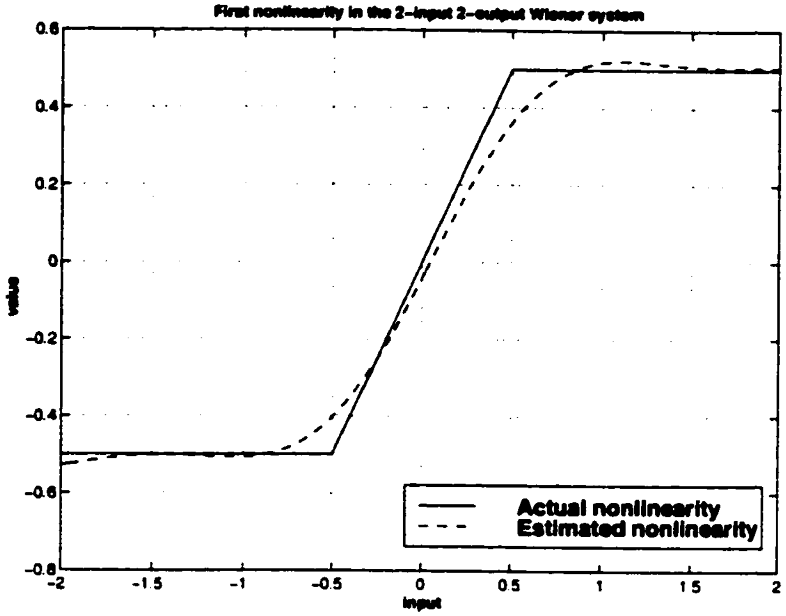


Figure 6.5: First nonlinearity in the 2-input 2-output Wiener system

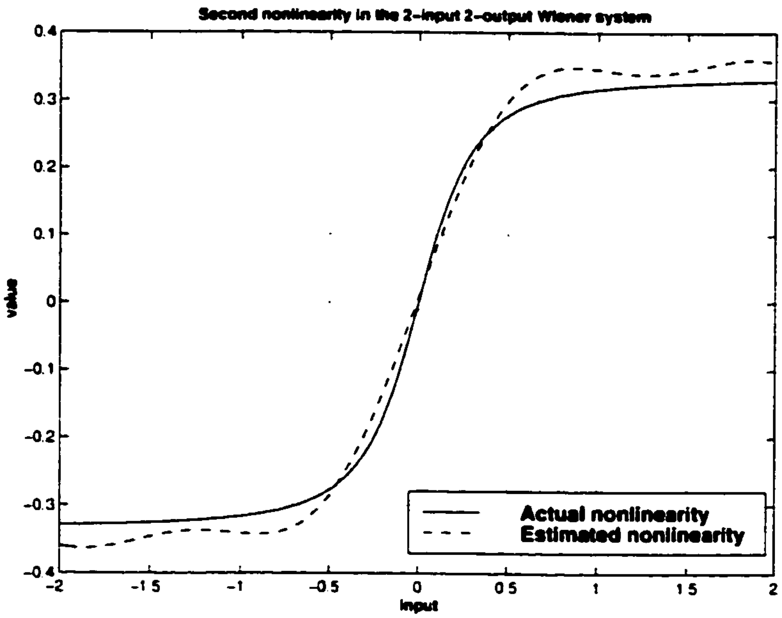


Figure 6.6: Second nonlinearity in the 2-input 2-output Wiener system

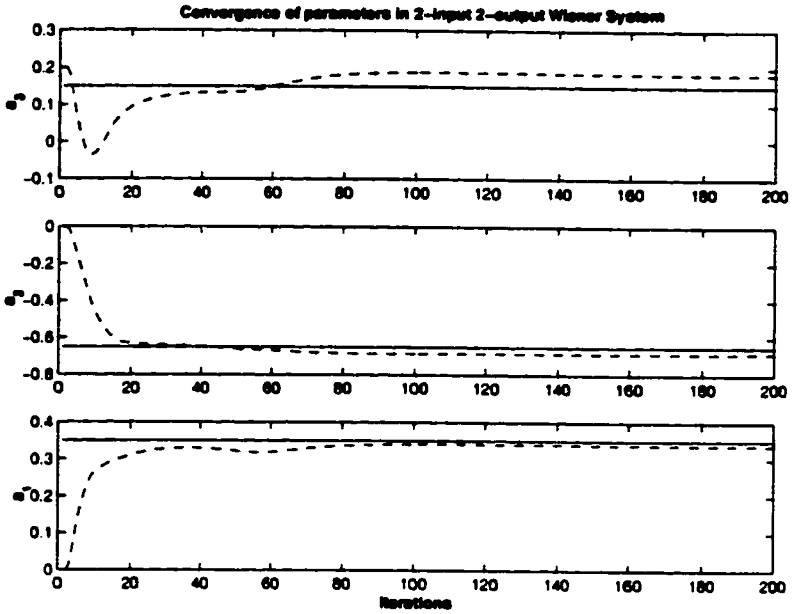


Figure 6.7: Convergence of poles of the linear part in 2-input 2-output Wiener system

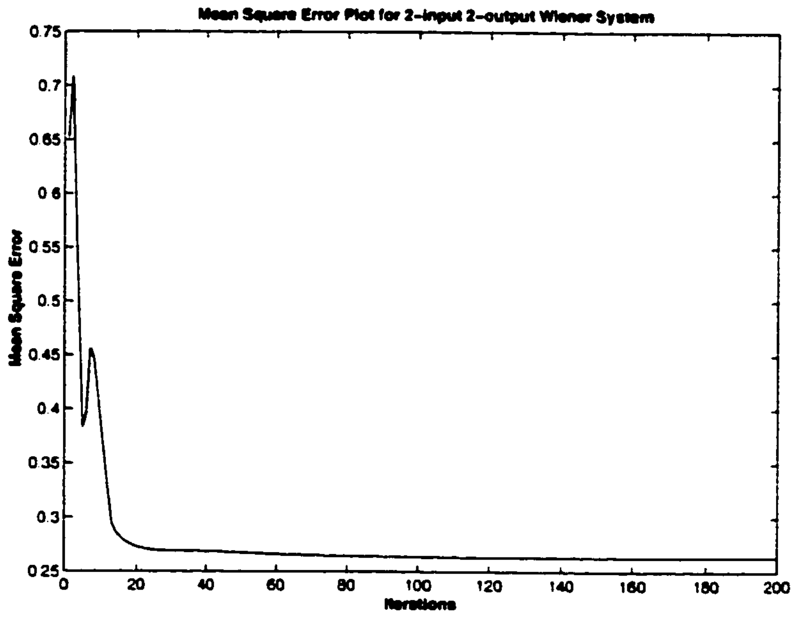


Figure 6.8: Square error plot for the 2-input 2-output system

# Chapter 7

# Conclusions and Future Work

This chapter concludes the thesis by summarizing important contributions and highlights some future avenues of research that can be originated from the work.

## 7.1 Conclusions

This thesis presents a new identification method structure for the identification of Hammerstein and Wiener systems. The new structure is composed of RBFNN/ARMA for Hammerstein and ARMA/RBFNN for Wiener type nonlinear systems. Training algorithms based on LMS approach have been developed for both the models. The update equations are derived for both SISO and MIMO cases. While considering the MIMO. systems with separate as well as combined nonlinearities are considered.

The proposed training algorithms are applied on several examples to validate the update equations. The results show excellent identification behaviour with very attractive rate of convergence, both in the sense of time and iterations. The hardware implementation of the proposed scheme, is also simpler due to lesser number of neurons needed, as compared to the number of neurons need in the MFNN approach. Moreover, lesser computations per iteration are required due to just one layer, and hence, time per iteration is also reduced. The proposed algorithm is also immune to noise as simulations show satisfactory identification behaviour even in a noisy environment.

## 7.2 Future Work

During the course of this thesis, it was found that, future research can be directed towards the following areas:

- Convergence and stability analysis of the proposed algorithms.
- This thesis has used a uniform distribution or trial selection of centers within the input space for the RBFNN. However, better means to locate the centers of the basis function are cited in the literature, *e.g.* $K$-means clustering.
- Faster convergence is expected by making the learning rate adaptive.
- This thesis considers nonlinear systems that can be modelled by Hammer-

stein and Wiener models. However, systems of Wiener-Hammerstein type or the Hammerstein-Wiener type can also be estimated using the same algorithm by adding another corresponding linear or nonlinear block.

# Nomenclature

### Abbreviations

| | |
|---|---|
| ARMA | Autoregressive Moving Average |
| LMS | Least Mean Squares |
| MFNN | Multi-layered Feedforward Neural Network |
| MIMO | Multi-input Multi-output |
| RBFNN | Radial Basis Functions Neural Network |
| SISO | Single-input Single-output |

### English Symbols

| | |
|---|---|
| $A(q^{-1})$ | The polynomial for the poles of the system |
| $a_i$ | The $i^{th}$ coefficient of the polynomial $A(q^{-1})$ corresponding to a delay $i$ in the output |
| $B^w_{jk}(q_{-1})$ | The polynomial of order $w$ for zeroes in the transfer function $G_{jk}$ |
| $b_{ijk}$ | The $i^{th}$ coefficient of the polynomial $B^w_{jk}(q_{-1})$ corresponding to a delay $i$ in the input |
| $C$ | The Centers of the basis functions |
| $c_i$ | Center of the $i^{th}$ basis function |
| $E$ | The error vector for MIMO system |
| $e(t)$ | The error for SISO system |
| $G_{jk}$ | Transfer function for the $j^{th}$ input and $k^{th}$ output |
| $I$ | Performance index |
| $K$ | The iteration number |
| $q^{-d}$ | The delay operator |
| $t$ | The time instant variable |
| $U(t)$ | The input vector for MIMO system |
| $u(t)$ | Input to the system |
| $u_m(t)$ | $n^{th}$ input for MIMO system |

| | |
|---|---|
| $W$ | The Weight vector |
| $w_i$ | weight corresponding to $i^{th}$ basis function |
| $X(t)$ | The intermediate variable vector for MIMO system |
| $x(t)$ | Intermediate nonmeasurable variable |
| $x_p(t)$ | $p^{th}$ Intermediate nonmeasurable variable for a MIMO system |
| $Y(t)$ | The output vector for MIMO system |
| $y(t)$ | Output of the system |
| $y_n(t)$ | $n^{th}$ output for a MIMO system |
| $\widehat{variable}(t)$ | Estimate of any variable |

**Greek Symbols**

| | |
|---|---|
| $\alpha$ | The learning rate |
| $\beta$ | Width of the basis function |
| $\phi(t)$ | The basis function |
| $\Phi(t)$ | The basis function vector |
| $\psi(t)$ | The regression vector |
| $\theta$ | The parameter vector |

# Bibliography


[1] V. Volterra. *Theory of Functions.* Blackie, 1930.

[2] H. L. Van Trees. *Synthesis of nonlinear optimal control systems.* MIT Press, 1962.

[3] J. F. Barret. Bibliography on volterra series, hermite functions expansions and realted subjects. *T. H. report 77-E-71, Eindhover University,* 1979.

[4] A. S. French and E. G. Butz. Measuring wiener kernels of a nonlinear systems using fft. *International Journal of Control,* 17:529–539, 1973.

[5] J. H. Seinfeld. Nonlinear estimation theory. *Ind. Eng. Chem.,* 62:32–42, 1970.

[6] S. A. Billings. Identification of non-linear systems - a survey. *IEE proceedings - part D,* 127(6):272–285, 1980.

[7] Q. Song and Y. C. Yin, L. Soh. Robust identification of nonlinear systems using neural network. 2000.

[8] L. Yong, H. Chongzhao, and D. Yingnong. Nonlinear system identification with genetic algorithms. 2000.

[9] Z. Zhu and H. Leung. Fuzzy based sliding manifolds for identification of a class of nonlinear systems. 2000.

[10] E.A. Wan and R. Van Der Merwe. *The IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000.*

[11] Gustavo Belforte and Paolo Gay. Hammerstein model identification with set membership errors. In *Proceedings of the* $38^{th}$ *conference on Decision and Control,* volume 1, pages 592–597, December 1999.

[12] Z. Zhu and H. Leung. Adaptive identification of nonliear systems with application to chaotic communications. 2000.

[13] V. Kadirkamanathan, M. H. Jaward, S. G. Fabri, and M. Kadirkamanathan. Particle filters for recursive model selection in linear and nonlinear system identification. 2000.

[14] Yong Fang and T. W. S. Chow. Orthogonal wavelet neural network applying to identification of wiener model. *IEEE transactions Circuit and Systems*, 47(4):591–593, 2000.

[15] Y. Li, X. Zhao, and L. Jioa. A nonlinear systems identification approach based on neuro-fuzzy networks. 2000.

[16] S. A. Billings and S. Y. Fakhouri. Identification of systems containing linear dynamic and nonlinear element. *Automatica*, 18:15–26, 1982.

[17] A. H. Haddad and J. B. Thomas. Optimal and suboptimal nonlinear filters for the discrete inputs. *IEEE transactions on Information Theory*, 35:409–418, 1968.

[18] E. DeBoer. Cross-correlation function of bandpass nonlinear networks. *Institute of Electrical and Electronic Engineers*, 16:1443–1446, 1976.

[19] J. H. Miller and J. B. Thomas. Detectors for the discrete time signalsin non-gaussian noise. *IEEE Transactions on Information Theory*, 18:241–250, 1972.

[20] T. E. McCannon, N. C. Gallagher, D. Mindo-Hamedani, and G.L. Wise. On the design of the nonlinear discrete time predictors. *IEEE Transactions on Information Theory*, 28:366–371, 1982.

[21] H. Al-Duwaish. A genetic appraoch to the identification of linear dynamical systems with static nonlinearities. *International Journal of Systems Science*, 31:307–313, 2000.

[22] M. Maqusi. Performance of baseband signals didgital transmission in nonlinear channel with memory. *IEEE Transactions on Communications*, 33:715–718, 1985.

[23] Anna Hagenblad. Identification of wiener models. In *First Conference on Computer Science and Systems Engineering in Linkping*, volume 1, pages 175–183. ECSEL, Jan 1998.

[24] E. Eskinat, S. H. Johnson, and W. L. Luyben. Use of hammerstein models in identification of nonlinear systems. *American Institute of Chemical Engineering Journal*, 37:255–268, 1991.

[25] I. W. Hunter and M. J. Korenberg. Identification of nonlinear biological systems: Weiner and hammerstein cascade models. *Biological Cybernetics*, 55:135–144, 1986.

[26] Kenneth J. Hunt, Marko Munih, Nick de Donaldson, and Fiona M. D. Barr. Investigation of hammerstein hypothesis in the modeling of electrically simulated muscle. *IEEE Transactions on Biomedical Engineering*, 45, 1998.

[27] J. N. Sandra, A. Palazaglu, and Romagnoli J. A. Application of wiener model predictive control (wmpc) to a ph neutralization experiment. *IEEE Transactions on Control Systems Technology*, 7:437–45, 1999.

[28] M. Singh, J. Elloy, r. Mezencev, and N. Munro. *Applied Indutsrial Control*. Pergamon Press, Oxford, UK, 1978.

[29] H. Al-Duwaish, M. Nazmul Karim, and V. Chandrasekar. Use of multilayer feedforward neural networks in identification and control of wiener model. *IEE Proceedings: Control Theory and Applications*, 143(3):225–258, 1995.

[30] E. Q. Doeblin. *Measurement Systems*. McGraw Hill Int., Auckland, New Zealand, 1983.

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

[32] F Chang and R Luus. A noniterative method for identification using hammerstein model. *IEEE Transactions on Automatic Control*, 16:464–468, 1971.

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

[34] W. Greblicki and M. Pawlak. Hammerstein system identification by nonparametric regression estmitation. *International journal of Control*, 45:343–354, 1987.

[35] Chyi Hwang and Kuo-Kai Shyu. Series expansion approach to the analysis and identification of discrete hammerstein systems. *International journal of Control*, 47:1961–1972, 1988.

[36] W. Greblicki. Non-parametric orthogonal series identification of hammerstein systems. *International Journal of Systems Science*, 20:2335–2367, 1989.

[37] A. H. Falkner. Identification of systems comprising of parallel hammerstein branches. *International Journal of Systems Science*, 47:2079–2087, 1988.

[38] Li zi qang. Identifiation of hammerstein and wiener systems using nonparametric estimation and best uniform approximation ploynomial. *IEEE Proceedings of the* $31^{st}$ *Conference on Decision & Control*, 3:933–934, 2000.

[39] W. Greblicki and M. Pawlak. Nonlinear system identification with nonparametric deconvolution. *IEEE Transactions on Information Theory*, page 124.

[40] M. Boutayeb and M. Darouch. Recursive identification method for the hammerstein model: Extension to nonlinear miso systems. *Control Theory and Advanced Technology*, 10:57–72, 1994.

[41] Sundeep Rangan, Greg Wolodkin, and Kameshwar Poolla. New results for hammerstein system identification. In *Proceedings of the* $34^{th}$ *conference on Decision and Control*, volume 1, pages 697–702. IEEE, December 1995.

[42] Jonathan C. Raltson and Abdelhak M. Zoubir. Identification of a class of mulitple input-output nonlinear systems driven by sationary non-gaussian processes. *IEEE transactions on signal processing*, 4:379–382, 1996.

[43] Michel Verhaegen and David Westwick. Identifying mimo hammerstein systems in the context of susbspace model indentification methods. *International Journal of Control*, 63:331–349, 1996.

[44] Adam Krzyzak and jerzy Z. Sasidek. A genetic appraoch to the identification of linear dynamical systems with static nonlinearities. *Conference*, 31:307–313, 2000.

[45] Sequare Daniel-Berhe and Heinz Unbehauen. Indentification of nonlinear continuous-time hammerstein model via hmf-method. December 1997.

[46] Yucai Zhu and Tai-Ji. Idetification of hammerstein models for control. *Proceedings of the 37th IEEE conference on Decision & Control*, 2000.

[47] H. X. Li. Idetification of hammerstein models using genetic algorithms. *IEE Proceedings Control theory Applications*, 146, 1999.

[48] Juan C. Gmez and Enrique Baeyens. Identification of multivariable hammerstein systems using rational orthonormal bases. In *Proceedings of the* $39^{th}$ *conference on Decision and Control*, volume 1, pages 2849–2854. IEEE. December 2000.

[49] Toshiharu Hatanaka and Katsuji Uosaki. Hammerstein model identification method based on genetic programming. *IEEE Instrumentation and measurement Technology conference*, pages 1430–1435.

[50] H. Al-Duwaish, M. Nazmul Karim, and V. Chandrasekar. Hammerstein model identification by multilayer feedforward neural networks. *International Journal of Systems Science*, 28(1):49–54, 1997.

[51] D. T. Westwick and R. E. Kearney. Identification of mutliple input wiener system. April 1990.

[52] D. W. Hu and Z. Z. Wang. An identifiaction method for the wieer model of nonlinear systems. Decmber 1991.

[53] Marco Lovera and miche Verhaegen. Recursive subspace identification of linear and nonlinear type wiener type models. september 1998.

[54] C. T. Chou and M. Verhaegen. Identification of wiener model with data prefiltering. December 1999.

[55] Anna Hagenblad and Lennart Ljung. Maximum likelihood estimation of wiener models. *IEEE Proceedings of the 39th Conference on Decision & Control*. 3:2417–2418, 2000.

[56] K. Rodriguez-Vázquez and P. J. Fleming. Multi-objective genetic programming for nonlinear system identification. April 1998.

[57] W. Greblicki. Nonparametric appraoch to wiener systems identification. *IEEE Transactions on Information Theory*, 44:538–545, 1997.

[58] W. Greblicki and M. Pawlak. Nonparametric identification of hammerstein systems. *IEEE Transactions on Information Theory*, 35:409–418, 1989.

[59] Wlodzmierz Greblicki. Continuous-time hammerstein system identification. *IEEE transactions on automatic control*. 45:1232–1236, 2000.

[60] Wlodzmierz Greblicki. Continuous-time wiener system identification. *IEEE transactions on automatic control*, 43:1488–1493, 1998.

[61] C. T. Chou and M. Verhaegen. Identification of wiener model with process noise. December 1999.

[62] Patrick Celka, Neil J. Bernard, and Jean-Marc Vesin. stochastic gradient identification of polymial wiener systems: Analysis and application. *IEEE transactions on signal processing*. 49:301–312, 2001.

[63] Anders E. Nordsj and Lars H. Zetterberg. Identification of certain time-varying wiener and hammerstein systems. *IEEE transactions on signal processing*, 49:577–592, 2001.

[64] Seth L. Lacy and R. Scott Erwin Dennis S. Bernstein. Identification of wiener systems with known nonivertible nonlinearities. June 2001.

[65] S. Haykin. *Neural Networks:A Comprehensive Foundation II.* Macmillan/IEEE Press, 1994, 1999.

[66] W. L. Jun-Dong, P. Jones. Comparison of neural network classifiers for nscat sea ice flag. 1998.

[67] L. Fortuna, G. Muscato, and M. G. Xibilia. A comparison between hmlp and hrbf for attitude control. 122:318–328, March 2001.

[68] R. A. Finan, A. T. Sapeluk, and R. I. Damper. Comparison of multilayer and radial basis functions neural networks for the text dependent speaker recognition. 1996.

[69] Hussain N. Al-Duwasih and S. Saad Azhar Ali. Identification of hammerstein model using radial basis functions neural networks. August 2001.

[70] T. Wigren. Recursive prediction error identification using the nonlinear wiener model. *Automatica*, 29:1011–1025, 1993.

[71] M. C. Kung and B. F. Womack. Discrete time adaptive control of linear dynamical systems with a two-segment piece-wise-linear symmetric nonlinearity. *IEEE Transaction on Automatic Control*, 29:173–180, 1984.

[72] H. Al-Duwaish and M. Nazmul Karim. A new method for the identifcation of hammerstein model. *Automatica*, 33:1871–1875, 1997.

[73] P. Gallman. An iterative method for the identification of nonlinear systems using uryson model. *IEEE transaction on Automatic Control*, 20:771–775, 1975.

[74] H. Al-Duwaish and W. Naeem. Nonlinear model predictive control using genetic algorithms. *IEEE Conference on Control Applications*, 20:771–775, 2001.

[75] Lang Zi-quang. Identification of hammerstein system using nonparametric estimation and best uniform approximation polynomial. In *Proceedings of the* $31^{st}$ *conference on Decision and Control*, volume 1, pages 933–934. IEEE, December 1992.

[76] G. Vandersteen and J. schoukens. Measurement and identification of nonlinear systems consisting out of linear and dynamic blocks and one static nonlinearity. *IEEE Instrumentation and measurement Technology conference*, pages 853–858.

[77] Philippe Crama and Johan Schoukens. First estimates of wiener systems using multisine excitation. *IEEE Instrumentation and measurement Technology conference*, pages 853–858.

[78] Wlodzimierz Greblicki. Nonlinearity estiamation in hammerstein systems based on ordered observation. *IEEE Transactions on signal processing*, pages 1224–1233.

[79] M. Boutayeb and M. Darouch. Identification of hammerstein model in the presennce of bounded disturbances. December 2000.

[80] Perre Alibert Marchi, Leanro dos Santos, and Aantonio Rodrigues Coelho. Comparative study of parametric and structural methodologies in identification of an experimental nonlinear process. August 1999.

[81] B. R. J. Haverkamp, C. T. Chou, and M. Verhaegen. Subspace identification of continuos-time wiener models. Decmber 1998.

[82] B. R. J. Haverkamp, C. T. Chou, and M. Verhaegen. Identifying mimo wiener systems using subspace model identification methods. Decmber 1995.

[83] Adam Krzyrk and Rolf Unehauen. On estimation of nonlinear systems by nonparametric techniques. *International Journal of Systems Science*, 31:189–192, 2000.

[84] Liamming Sun, Wenjiang Liu, and Akira Sano. Least squares identification of hammerstein model based on over-sampling scheme. *UKACC International Conference on Control*, 1996.

[85] W. Greblicki and M. Pawlak. Identification of discrete hammerstein system using kernel regression estimates. *IEEE transactions on Automatic Control*, 31:74–77, 1986.

[86] D. W. Hu and Wang Z. Z. An identification method of wiener model of nonlinear systems. *IEEE Proceedings of the 30th Conference on Decision Control*. 37:783–786, 1991.

[87] S. Hassouna, P. Coirault, and R. Ouvrard. *Proceedings of the American Control Conference, 2001.*

[88] Li Yong, Han Chongzhao, and Dang Yingnong. Nonlinear system identification with genetic algorithms. *Proceedings of the 3rd World Congress on Intelligent Control and Automation, 2000.*, 1(1):597–601, 2000.

[89] Visala A. Maximum likelihood estimation of wiener models. *IEEE World Congress on Computational Intelligence*, 2:1281 –1286, 1998.

[90] B.R.J. Haverkamp, C.T. Chou, and U Verhaegen. Maximum likelihood estimation of wiener models. *Proceedings of the 37th IEEE Conference on Decision and Control*, 2:1846 –1847, 1998.

[91] C.T. Chou and U Verhaegen. An indirect approach to closed loop identification of wiener models. *Proceedings of American Control Conference*, 5:3451 –3455, 1999.

# Vita

- Syed Saad Azhar Ali.
- Born in Jeddah, Saudi Arabia on December 10, 1975.
- Received Bachelor of Engineering (B.E) degree in Electrical Engineering from N.E.D University of Engineering and Technology, Karachi, Pakistan in 1999.
- Joined King Fahd University of Petroleum and Minerals in January 2000.
- Publication: Hussain, N. Al-Duwaish and Ali, Syed Saad Azhar, 'Hammerstein Model Identification Using Radial Basis Functions Neural Networks', *International Conference of Artificial Neural Networks (Icann), August 2001, Vol 1, pp: 951-956.*
- *Email:* <u>*saadi@ieee.org*</u>