U-Model Based Adaptive Internal Model Control for Tracking of Nonlinear Dynamic Plants

BY

Naveed Razzaq Butt

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

Systems Engineering

December 2005

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

DEANSHIP OF GRADUATE STUDIES

This thesis, written by NAVEED RAZZAQ BUTT 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 SYSTEMS ENGINEERING.

THESIS COMMITTEE

Dr. Muhammad Shafiq (Advisor)

Dr. Fouad. M. Al-Sunni (Member)

Dr. Moustafa. A. El-Shafei (Member)

Dr. Umar M. Al-Turki
Department Chairman

Dr. Mohammad A. Al-Ohali
Dean of Graduate Studies

May 15, 2006
Date

To my beloved family;

who always told me I could do it!!

ACKNOWLEDGEMENTS

All praise and glory is to Allah Almighty who taught man through the pen. And, peace and blessing of Allah be on His last messenger, Muhammad (Sallallah-Alaihe-Wasallam), who brought the absolute truth and wisdom to mankind.

This thesis owes its existence to the King Fahd University of Petroleum & Minerals (KFUPM) and a number of persons both inside and outside the university. Thanking all of them individually may not be possible here, but I would like to name a few.

First and the foremost acknowledgement is due to the Department of Systems Engineering that gave me the opportunity to work on my master's program in the capacity of a Research Assistant. I am also indebted to KFUPM for supporting my research work and providing a genial research environment.

My deep appreciation and heartfelt gratitude goes to my thesis advisor Dr. M. Shafiq for his constant endeavor, guidance and encouragement throughout the course of this research work. Perhaps the most helpful of all were his enthusiastic approach towards research and his ready-availability at all times. His valuable suggestions made this work interesting and knowledgeable for me. Working with him in a friendly and motivating environment was really a joyful and learning experience that is not easy to forget.

I extend my deepest gratitude to Professor F. Al-Sunni and Dr. M. El-Shafei for accepting to be in my thesis committee. Their constructive and positive criticism and thought-provoking suggestions were extremely helpful. It was surely an honor to have them in the thesis committee.

Special thanks are also due to the department chairman Dr. U. Al-Turki who on multiple occasions reduced my departmental workload, thus allowing me to concentrate on my research work.

I also owe thanks to a number of other persons whose presence made my experience at KFUPM both amusing and rewarding. These include Aijaz, Tahir, Salman,Tayyab Bhai, Basharat Bhai, Maj. Amir, Rami, Suraj, Osama, Fahd and Ather Bhai. Additionally, Shahzada Basharat's Latex template for this thesis and Aijaz's formatting of the equations made my life very easy and for that I am eternally grateful.

Last but not least, thanks are due to my parents and rest of the family members who always prayed for my success and without whose continuous encouragement and unwavering support I would never have made it this far!

TABLE OF CONTENTS

LIST OF FIGURES

THESIS ABSTRACT

Name: Naveed Razzaq Butt

Title: U-Model Based Adaptive Internal Model Control for Tracking of Nonlinear Dynamic Plants

Major Field: Systems Engineering

Date of Degree: December 2005

Adaptive tracking of nonlinear dynamic plants is currently an important area of research. The main difficulty being felt by the research community is the lack of a general modelling framework that can facilitate synthesis of a simplistic control law, while being capable of providing accurate approximation of nonlinear systems. The aim of this study is to alleviate that problem by introducing a novel technique based on the control-oriented U-Model for the adaptive tracking of a wide range of stable nonlinear dynamic plants using only input-output data. The overall scheme is based on the robust internal model control (IMC) structure wherein different internal models, using nonlinear adaptive filtering and higher-order neural networks, are proposed. In each case, the U-Model equivalence of the internal model is developed and a simplistic control law based on polynomial root-solving is synthesized. Stability of the proposed adaptive scheme is analyzed with the help of various analytical tools. The effectiveness of the proposed adaptive schemes is demonstrated through simulations and real-time applications to a variety of nonlinear plants that include: Hammerstein model, CSTR model, DC motor and single-link robotic arm manipulator.

CHAPTER 1

INTRODUCTION

1.1 Overview

Recent advances in a variety of technologies and applications call for improved performance and reliability, while exacerbating the complexity and uncertainty of systems and their surroundings. In many instances, the operation of systems and devices can be modified and, possibly, optimized by the intervention of a control system. The objective of the control is to influence the behavior of systems. One such objective is *tracking* which involves forcing system variables to follow prescribed trajectories.

On the design side, some of the main difficulties to be overcome by the designer are:

1. Nonlinear plant dynamics,

2. Uncertainties caused by differences between actual and assumed dynamic models

It is, therefore, not surprising that these two considerations form the dominant areas of interest in modern control design. Most physical systems are nonlinear and multivariate. By nature, they have inherent interconnected nonlinearities in their dynamics where the relationship between the input and output variables varies depending on the operating conditions. Many common control problems involve dynamic systems that exhibit nonlinear behavior. If the nonlinearities are mild or the operating conditions do not change much, then the effect of the nonlinearities is not severe, and linear control techniques are applicable. This has been the approach of researchers in the past many decades and has resulted in the production of a huge amount of literature in the areas of linearization and linear control [1–3].

However, many industrial systems exhibit strong nonlinear behavior and they may be required to operate over a wide range of operating conditions. Additionally, there are situations where the nonlinear plants are extremely difficult to model and/or they exhibit high uncertainties. Under these conditions, the conventional constant gain feedback controller fails to maintain the performance of the system at acceptable levels and does not respond well to changes in system dynamics. This has resulted in an increased interest in developing controllers whose parameters can vary online (thus *adapting* to the current plant dynamics). Adaptive tracking of uncertain nonlinear dynamic plants has become an important area of research and several adaptive control designs have been recently developed. These include, among others, Self Tuning (ST) Control, Model-Based Control (MBC), Neural Control and Fuzzy Control for both linear and nonlinear systems. An extensive discussion on most of these designs can be found in [4].

In particular, MBC has received great attention by designers of nonlinear adaptive systems [5]. This is due to a number of reasons. Firstly, there exist a good number of model-based control structures for use in a variety of situations, and can be extended to nonlinear design rather conveniently. Secondly, MBC schemes can benefit from a wide range of tools available for modelling and estimation. Thus one can find MBC schemes utilizing tools from adaptive filtering [6], neural networks [7, 8] and fuzzy logic [9]. MBC provides, in a way, a platform for combining and testing of a variety of modelling, identification and controller-design methodologies.

Nonlinear Model-Based Control (NMBC) strategies generally involve three important steps:

1. Selection of a suitable overall structure
2. Selection of a proper modelling framework
3. Synthesis of a control law based on the two choices above

For the first choice, there are a good number of structures available in literature (see [5,10]). Among them, Internal Model Control (IMC) has become an industry favorite because of its robustness and disturbance rejection capabilities [11]. A nonadaptive version of the IMC requires an explicit model of the plant and a stable controller(which is supposed to be the inverse of the model). Since the model used can be linear or nonlinear, IMC provides possibility of extension to nonlinear control [12]. However, since finding the inverse of nonlinear models is much more involved then simple linear models, the nonlinear extension of the IMC raises a number of performance issues. The good part is that, once the issue of finding the nonlinear model inverse is aptly solved, the IMC guarantees excellent tracking performance.

Additionally, although the IMC requires an input-output stable plant, this does not limit its general utility. This is because of two reasons. Firstly, most industrial systems happen to be inherently open-loop stable. Secondly, most unstable plants can be stabilized using simple feedback and can then be placed in the IMC structure for robust tracking.

Further extension of the nonlinear IMC to include adaptive tracking of uncertain nonlinear plants poses a number of challenges [7, 13]. These include the need to have suitable modelling framework, the ability to rapidly identify the plant dynamics and the computational power to obtain the nonlinear model inverse (to be used as the controller), whilst assuring overall stability. Among these, perhaps the most crucial factor is the nature of the model used. This has been mentioned as the second important choice for NMBC schemes in the list above and needs to be discussed further.

It is clear that the modelling part plays a very important role in the overall system design and performance. This is because the control law is actually based upon the model structure. Two of the more important measures of model utility are its approximation capabilities (in terms of accuracy) and suitability for control (in terms of computational ease) [14]. Unfortunately, the research community so far feels a complete nonexistence of a general modelling framework that satisfies both of the above mentioned measures simultaneously [14–17]. For instance, linear models, that are easy to manoeuver, are inherently incapable of describing an enormous range of important dynamic phenomena. On the other end of the spectrum, we take the most commonly used nonlinear modelling structure; the NARMAX (Nonlinear Autoregressive Moving Average with Exogenous inputs) Model. It yields the input-output representation of nonlinear systems by defining the current output using a nonlinear functional expansion of lagged inputs, outputs and noise terms [18]. Thus depending on representation of the functional expansion and its parametrization, different model structures such as the polynomial NARMAX can be defined. Moreover, many well-known nonlinear models (e.g., Hammerstein, Wiener, Bilinear, Nonlinear-FIR etc.) can be shown to form special classes of the NARMAX model [14, 18]. Thus the

NARMAX is an excellent tool for representing nonlinear systems. However because of its overly complex structure(a huge number of parameters are needed to characterize even simple nonlinear systems), it does not lend itself to easy manoeuvring for controller design [19].

As outlined in the discussion above, there exists a genuine need for a model based tracking scheme for nonlinear plants that is general, sufficiently accurate and enables the synthesis of a simplistic control law. That is where the work proposed in this thesis comes in. We develop an adaptive nonlinear scheme that is robust and leads to computationally and analytically simplistic control law.

The key idea is based on the recently proposed control-oriented nonlinear model termed as the U-model [17]. The U-Model expands the nonlinear NARMAX function as power series in the current control term, thus allowing simple polynomial root-solving procedures to be used for controller synthesis. Additionally, the U-Model has a more general appeal as compared to the polynomial NARMAX model [20] and Hammerstein model. Based on the U-Model a pole-placement controller [17] and a new IMC structure for dynamic nonlinear plants with known parameters [21] have recently been proposed.

In this thesis, we introduce an *adaptive* IMC scheme based on the U-Model utilizing nonlinear adaptive filtering and neural networks. Our main objective is to combine the robustness of the IMC and the control-oriented nature of the U-Model with the approximation capabilities of nonlinear adaptive filters and neural networks to provide a comprehensive nonlinear adaptive control scheme.

Next, we present a formal problem statement followed by thesis contributions and organization.

1.2 Problem Statement

The problem to be studied in this work can be formalized as follows:

> *"Given a stable nonlinear plant with uncertain or un-modelled dynamics, design and implement a U-Model based IMC structure to achieve adaptive tracking"*

1.3 Objectives Achieved

1. A U-Model based adaptive IMC structure is proposed for the adaptive tracking of stable single-input-single-output(SISO) nonlinear dynamic plants.
2. Three different nonlinear models are proposed for model identification in the adaptive IMC. These include:
   (a) A Radial Bases Functions (RBF) based nonlinear model
   (b) A new general polynomial-kind nonlinear model
   (c) A Higher-Order Neural Network (HONN) based nonlinear model
3. For each of the modelling frameworks above, a U-Model equivalence is developed and a controller is synthesized.
4. The use of normalized Leaky Least Mean Square algorithm (nLLMS), for the model identification, is proposed and justified in detail.
5. Stability analysis of the proposed adaptive IMC scheme is carried out using the small gain theorem and internal stability.
6. The use of Newton-Raphson algorithm for controller synthesis in the proposed adaptive IMC is studied and justifications for its applicability are provided.

7. Simulations (using SIMULINK) are carried out for the adaptive tracking of a number of nonlinear dynamic plants using the proposed work. These plants include:
    (a) Hammerstein Model
    (b) Continuously Stirred Tank Reactor (CSTR) model
    (c) Permanent Magnet DC-Motor Model
8. Real-time implementation of the proposed adaptive IMC to the adaptive tracking of laboratory-scale DC-Motor speed using SIMULINK platform.
9. Real-time implementation of the proposed adaptive IMC to the adaptive tracking of two nonlinear plants that are initially unstable. The two plants are:
    (a) Permanent Magnet DC-Motor position tracking
    (b) Single-link robotic arm manipulator position control

1.4 Thesis Organization

The rest of the thesis has been organized as follows: Chapter 2 provides an introductory treatise to the area of nonlinear adaptive control. The topics discussed here, in brief, include: nonlinear systems, common nonlinearities, modelling of nonlinear systems, adaptive and intelligent control of nonlinear systems. This is followed by a review of the literature in these areas in chapter 3.

Chapter 4 covers the proposed adaptive IMC structure using RBF based nonlinear model. Detailed discussions on related topics are provided. These include: properties of the nonlinear IMC, the RBF based nonlinear model and algorithm for model updating (nLLMS). Additionally, stability analysis of the proposed scheme is carried out in the same chapter. Simulations and real-time implementation results are also presented. A new nonlinear adaptive filter based model for use with the adaptive IMC is proposed in chapter 5. Its U-Model equivalence is developed and a Newton-Raphson algorithm based controller is synthesized. The effectiveness of this proposed scheme is demonstrated through simulations and real-time implementation. Chapter 6 proposes the use of HONNs in the U-Model based adaptive IMC scheme. U-Model equivalence of the Higher-Order Neural Unit (HONU) is developed and an iterative root-solving controller is synthesized. Again, simulation results demonstrating the effectiveness of the proposed scheme are presented.

Finally, conclusions and suggestion for future work are presented in chapter 7.

CHAPTER 2

NONLINEAR ADAPTIVE CONTROL: AN INTRODUCTORY FRAMEWORK

2.1 Nonlinearity

In the linear world, the relation between cause and effect is constant and the relation is quite independent of magnitude. For instance, if a force of 1 newton, applied to a mass m, causes the mass to accelerate at a rate a, then according to a linear model, a force of 100 newtons, applied to the same mass, will produce an acceleration of $100a$. Strictly a linear function f must satisfy the following two condition, where α is a scalar:

$$\left.\begin{aligned} f(u_1(t)) + f(u_2(t)) &= f(u_1(t) + u_2(t)) \\ f(\alpha u_1(t)) &= \alpha f(u_1(t)) \end{aligned}\right\} \tag{2.1}$$

Any system whose input-output characteristic does not satisfy the above conditions is classified as a nonlinear system. Thus, there is no unifying feature present in nonlinear systems except the absence of linearity. Nonlinear systems sometimes may not be capable of analytical description; they may sometimes be discontinuous or the may contain well understood smooth mathematical functions [3].

2.1.1 Inapplicability of Linear Tools

The following statements are broadly true for nonlinear systems:

1. Matrix and vector methods, transform methods, block-diagram algebra, frequency responde methods, poles and zeros and root loci are all inapplicable.

2. Available methods of analysis are concerned almost entirely with providing limited stability information.

3. System design/synthesis methods are not abundant.

4. Numerical simulation of nonlinear systems may yield results that are misleading or at least difficult to interpret. This is because, in general, behavior of a nonlinear system is structurally different in different regions of the state space.

2.1.2 State-Space Representation

The input u, output y and system state x are related through the nonlinear functions $f(.)$ and $g(.)$ as:

$$\dot{x} = f(x, u); \quad y = g(x); \quad x \epsilon X \tag{2.2}$$

2.2 Nonlinear Control

The subject of nonlinear control deals with the analysis and design of nonlinear control systems, i.e., of control systems containing at least one nonlinear component. In the design, we are given a nonlinear plant to be controlled and our task is to construct a controller so that the closed-loop system meets the desired characteristics.

2.2.1 Need for Nonlinear Control

Since linear control is a mature subject with a variety of powerful methods a natural question might arise: why use nonlinear control? Among a number of reasons for using nonlinear control, a few are given here [22]:

1. Improvement of existing control systems: Linear control methods rely on the key assumption of small range operation for the linear model to be valid. Thus a linear controller may not perform well or may become unstable when the operating range is large. Nonlinear controllers, on the other hand, may handle the nonlinearities in the large range operation directly.

2. Analysis of hard nonlinearities: Systems exhibiting hard nonlinearities such as Coulomb friction, saturation, dead-zones, backlash and hysteresis are often found in engineering applications. Their effects cannot be derived from linear methods and therefore, nonlinear methods must be used.

3. Dealing with model uncertainties: Linear control schemes usually assume that the parameters of the system model are reasonably well known. However, many control problems involve uncertainties in model parameters. Nonlinearities can be intentionally introduced in the controller part so that model uncertainties can be tolerated. Two classes of nonlinear controllers for this purpose are robust controllers and adaptive controllers.

2.2.2 Commonly Used Nonlinear Control Methods

Among the various available nonlinear control methods, the most widely used ones are:

1. Feedback Linearization Of Nonlinear Systems

2. Nonlinear Output Regulation

3. Lyapunov Design

4. Sliding-Model Controller Design

5. Adaptive Nonlinear Control

6. Neural Control

7. Fuzzy Control

The work in this thesis is related to items 5 and 6. Adaptive nonlinear control using IMC is discussed in detail in chapter 4, while higher-order neural networks based IMC is presented in chapter 6.

2.3 Nonlinear Adaptive Control

Although research in the area of adaptive control began in early 1950's, it was only in the last twenty five years that a coherent theory of adaptive control was developed, using various tools from nonlinear control theory. These theoretical advances along with the availability of cheap computation, have lead to many practical applications, in areas such as robotic manipulation, aircraft and rocket control, chemical processes, power system, ship steering and bioengineering.

2.3.1 Need for Adaptive Control

Among a number of reasons, the most significant few reasons and motivations for using adaptive control are mentioned below:

1. Many control tasks, such as robot manipulation, involve parameter uncertainty at the beginning of the control operation. Unless such parameter uncertainty is gradually reduced online by an adaptation mechanism, the system may fail to perform.

2. For many other tasks, such as those in power system, the system dynamics may be well known at the beginning, but these may exhibit unpredictable parameter variations as the control operation goes on. Thus without a continuous redesign of the controller, the initially suitable controller may not be able to control the system well

3. In many other situations, it might be extremely difficult to model the plant using first principle and only input-output relations might be available. In such cases, control schemes that can identify the system online and make it track certain trajectories are essential.

2.3.2 Nonlinear Model Based Control: NMBC

There are a number of adaptive control schemes available for use. Among these, model-based (or indirect) control schemes have gained great importance in industrial applications. Nearly all model-based control schemes can be represented by the block diagram shown below (figure 2.1). The system essentially contains the nonlinear plant to be controlled, a model of the plant and the controller. An adaptation algorithm updates the model based on the observation of the model and plant inputs and outputs. The controller design block then computes the controller function, treating the model as if it were the actual plant (this approach is commonly known in literature as the *certainty equivalence principle*). The IMC structure, used in this work, is a special case of model-based controllers and is discussed in detail in chapter 4.

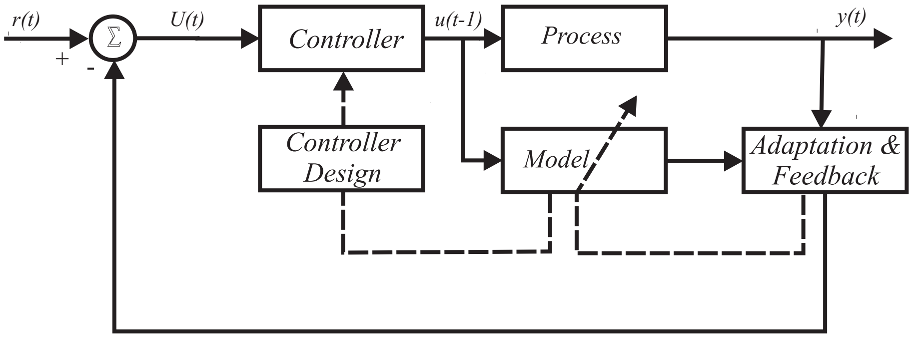

Figure 2.1: General MBC Schematic

2.4 Modelling of Nonlinear Systems

Modelling of nonlinear systems plays a very important role in the design and analysis of adaptive nonlinear control schemes such as the NMBC. Unfortunately, most modern surveys on modelling of nonlinear systems (e.g., [14–17]) come to the same conclusion: there is no systematic approach for building nonlinear models for NMBC. As pointed out in the introduction (chapter 1), the main difficulty in the area of nonlinear control is the lack of a general modelling framework that allows the synthesis of a simplistic control law while being sufficiently accurate. Since the work in this thesis is based on the newly developed "U-Model" that aims at alleviating this problem, we present here the commonly available classes of nonlinear models and then go on to discuss the details of the U-Model.

2.4.1 NARMAX and NARMAX-Based Models

NARMAX (Nonlinear Autoregressive Moving Average with Exogenous inputs), the most commonly used nonlinear model has the ability to represent a large number of nonlinear systems and covers several of the other nonlinear modelling structures as its sub-classes. Therefore in this section we present the NARMAX model and the other commonly used nonlinear models that form special cases of the NARMAX model. For the sake of brevity, in many cases, only the most commonly used cases of the models are given and their NARMAX equivalence is not presented separately. A detailed discussion on most of these models can be found in [14].

In the following u_i, y_i and e_i represent the input, the output and the error. Nonlinear functions are designated by $f(.)$ and $g(.)$ and constants by a_i, b_i and c_{ij}. Any other variables used are defined separately with each model.

- **NARMAX:**

$$y_k = f(y_{k-1}, \ldots, y_{k-p}, u_k, \ldots, u_{k-q}, e_{k-1}, \ldots, e_{k-r}) + e_k \tag{2.3}$$

Here $f(.)$ is a nonlinear function of the p past outputs, the current and q past inputs and the past r elements of the prediction error sequence.

- **Polynomial NARMAX:** A particular parameterization of the NARMAX model is to expand its nonlinear function $f(.)$ as a polynomial of order L w.r.t. lagged inputs, outputs and error terms. This leads to:

$$y_k = a_1 + \sum_{i=2}^{N} a_i \prod_{j=1}^{P_i} y_{k-p_j} \prod_{l=1}^{Q_i} u_{k-q_l} \prod_{m=1}^{R_i} e_{k-r_m} + e_k \tag{2.4}$$

with $1 \leq P_i + Q_i + R_i \leq L$ for all values of i.

- **Nonlinear FIR Models:**

$$y_k = f(u_{k-d}, \ldots, u_{k-q}) \tag{2.5}$$

where $q \geq d \geq 0$. This modelling structure also covers the nonrecurrent neural network models.

- **Finite Volterra Models:**

$$\left.\begin{aligned} y_k &= \alpha_0 + \sum_{i=1}^{N} {v^n}_M(k) \\ {v^n}_M(k) &= \sum_{i_1=0}^{M} \ldots \sum_{i_n=0}^{M} \alpha_n(i_1, \ldots, i_n) u_{k-i_1} \ldots u_{k-i_n} \end{aligned}\right\} \tag{2.6}$$

where α_j represent piecewise continuous functions.

- **Hammerstein Model:**

$$y_k = \sum_{i=1}^{P} a_i y_{k-i} + \sum_{i=0}^{q} b_i g(u_{k-i}) \tag{2.7}$$

- **Wiener Model:**

$$\left.\begin{aligned} w_k &= \sum_{i=1}^{P} a_i w_{k-i} + \sum_{i=0}^{q} b_i u_{k-i} \\ y_k &= g(w_k) \end{aligned}\right\} \tag{2.8}$$

- **Lur'e Models:**

$$y_k = \sum_{i=1}^{P} a_i y_{k-i} + \sum_{i=1}^{q} b_i [u_{k-i} - g(y_{k-i})] \tag{2.9}$$

- **NARX:**

$$y_k = f(y_{k-1}, \ldots, y_{k-p}) + b_0 u_k \tag{2.10}$$

- **Output-Affine Models:**

$$y_k = \sum_{i=1}^{P} \alpha_i(\mathbf{u}_k) y_{k-i} + \beta(\mathbf{u}_k) \tag{2.11}$$

where, vector $\mathbf{u}_k = [u_k, \ldots, u_{k-q}]$ while $\alpha_i(.)$ and $\beta(.)$ represent arbitrary non-linear functions (usually polynomials).

- **Bilinear Input-Output Models:**

$$y_k = \sum_{i=1}^{p} a_i y_{k-i} + \sum_{i=1}^{q} b_i u_{k-i} + \sum_{i=1}^{P} \sum_{j=1}^{Q} c_{ij} y_{k-i} u_{k-j} \tag{2.12}$$

2.4.2 The U-Model

The control-oriented U-Model plays a central role in the adaptive scheme proposed in this paper. Following is the development of the U-model based on [17]. Consider single-input single-output (SISO) nonlinear dynamic plant with a NARMAX representation of the form:

$$y(t) = f[y(t-1), \ldots, y(t-n), u(t-1), \ldots, u(t-n), e(t), \ldots, e(t-n)] \tag{2.13}$$

where $y(t)$ and $u(t)$ and are the output and input signals of the plant respectively at discrete time instant t, n is the order of the plant, $f(.)$ is a nonlinear function and $e(t)$ represents the error due to measurement noise, model mismatch, uncertain dynamics and plant variation. The U-Model is obtained by expanding the nonlinear function $f(.)$ of the above equation as a polynomial with respect to $u(t-1)$ as follows:

$$y(t) = \sum_{j=0}^{M} \alpha_j(t) u^j(t-1) + e(t) \tag{2.14}$$

where M is the degree of model input $u(t-1)$, $\alpha_j(t)$ is a function of past inputs and outputs $u(t-2), \ldots, u(t-n), y(t-1), \ldots, y(t-n)$ and errors $e(t-1), \ldots, e(t-n)$ To apply linear control system design methodologies to the nonlinear model a further transformation is applied as follows:

$$y(t) = U(t) \tag{2.15}$$

where,

$$U(t) = \Phi[u(t-1)] + e(t) = \sum_{j=0}^{M} \alpha_j(t) u^j(t-1) + e(t)$$

The expression (2.15) is defined as the U-Model. This model has the following advantages:

1. The control-oriented U-Model is more general than other parameterizing approaches, such as the polynomial NARMAX model [20], the Hammerstein model etc.

2. The sampled data representation of many nonlinear continuous time systems can be of the form
$$y(t) = \sum_{j=0}^{M} \alpha_j(t) u^j(t-1)$$

3. The U-model exhibits a polynomial structure in the current control $u(t-1)$.

4. Due to its polynomial structure, the nonlinear algebraic equations, which need to be solved to obtain the output value of the controller, are also polynomials in $u(t-1)$, unlike other models which lead to complex nonlinear algebraic equations.

CHAPTER 3

LITERATURE REVIEW

3.1 Introduction

We now present a comprehensive review of the research in the area of adaptive nonlinear control. We will concentrate mainly on nonlinear control strategies using the U-Model and IMC. However, brief coverage of general nonlinear control and neural networks is also included for completion. Since the amount of literature in the area is overwhelmingly large, we try to present only typical works.

3.2 U-Model Based Control

Zhu and Guo [17], in 2002, introduced a new control oriented model called the U-Model. The U-Model is designed to make the control synthesis part (of the nonlinear control) simpler. It was shown to have a more general appeal as compared to the polynomial NARMAX model and Hammerstein model. They also developed a pole-placement controller based on the U-Model.

In 2004, Shafiq and Haseeb [21] developed a U-Model based IMC structure for the control of known stable nonlinear dynamic plants. The structure was shown to be robust and capable of covering a large variety of nonlinear processes.

3.3 IMC Based Nonlinear Adaptive Control

In 1982, Garcia and Morari [11] first defined the internal model control (IMC) structure for single-input single-output (SISO), discrete-time systems. Several new stability theorems for IMC were proved and it was concluded that the IMC structure allows a rational design procedure where, in the first step, the controller is selected to give perfect control. In the second step, a filter is introduced which makes the system robust to model-plant mismatches.

Economou et al. [12], in 1986, presented the nonlinear version of the IMC and discussed its properties in detail. A controller was developed based on numerical methods and the overall stability of the system was discussed.

Hunt and Sbarbaro [7], in 1991, proposed a novel technique of directly using artificial neural networks for the adaptive control of nonlinear systems. The use of nonlinear function inverses was investigated and IMC was used as the control structure.

In a further contribution [8], in 1993, they presented artificial neural network architectures for the implementation of nonlinear IMC. This approach can be viewed as a nonlinear analogue of adaptive inverse control; the network models used were equivalent to nonlinear adaptive filters. They used two separate networks in the implementation of nonlinear IMC; one network models the plant, and the second network models the plant inverse.

In 1996, Datta and Ochoa [13] combined adaptation with an internal model control structure to obtain an adaptive IMC scheme possessing theoretically provable guarantees of stability. The adaptive IMC scheme was designed for open-loop stable plants using the traditional certainty equivalence approach of adaptive control and it was shown that using a series-parallel identification model, for a stable plant, one can adapt the internal model on-line and guarantee stability and asymptotic performance in the ideal case.

Sousa et al. [9], in 1997, presented NMBC based on a Takagi-Sugeno fuzzy model. The controller proposed is a combination of a model predictive controller and an inverse based controller (IMC). If no constraints are violated, an inverse based control algorithm is used. When constraints are violated, the required optimization used is a branch-and-bound algorithm. They illustrate their controller by applying it to a laboratory scale air-conditioning system.

Lightbody and Irwin [23] investigated in detail the possible application of neural networks to the modelling and adaptive control of nonlinear systems. Nonlinear neural-network-based plant modelling was discussed, based on the approximation capabilities of the multilayer perceptron. A novel nonlinear IMC strategy was suggested, that utilizes a nonlinear neural model of the plant to generate parameter estimates over the nonlinear operating region for an adaptive linear internal model, without the problems associated with recursive parameter identification algorithms. Unlike other neural IMC approaches the linear control law can then be readily designed. The proposed schemes was applied to the tracking of the CSTR plant.

Choi and Kim [24] proposed a robust adaptive controller based on the IMC structure for stable plants. A stable high order model for the stable plants using the RLS algorithm and its stable reduced order model is calculated using the ordered real Schur form method. The stable adaptive IMC controller is designed for the reduced order model and is augmented by the low-pass filter such that the closed loop stability for the higher order model is ensured.

Kambhampati et al. [25] presented several theoretical results for the application of recurrent neural networks to the production of an IMC system for nonlinear plants. The results include determination of the relative order of a recurrent neural network and invertibility of such a network. A closed loop controller was produced without the need to retrain the neural network plant model and the stability of the closed-loop controller was demonstrated.

The 1999 tutorial by Hagan and Demuth [26], provides a quick overview of neural networks and explains how they can be used in control systems. Included in the tutorial are several control architectures, such as model reference adaptive control, model predictive control, and internal model control.

A brief summary of most recent works (years 2000 to 2005) in the area of nonlinear internal model control is now presented.

Rivals and Personnaz [27] presented a new nonlinear IMC using neural networks for control of processes with delay. The internal model used is a delay-deprived model cascaded with fixed delay, while the inverse model is based on neural network models and consists of the inverse of the delay-deprived model only.

Xie and Rad [28] presented a fuzzy adaptive internal model controller for open-loop stable plants. The control scheme consists of a dynamic model and a model-based fuzzy controller. Fuzzy dynamic model which serves as the internal model is identified online by using the input and output measurement of the plant. Based on the identified fuzzy model, a fuzzy controller is designed.

Fink and Nelles [29] extended the IMC scheme to nonlinear processes based on local linear models where the properties of the linear design procedures can be exploited directly. A local linear neuro-fuzzy model is used. The output of the model is calculated as an interpolation of locally valid linear models. Local model architectures allow the separate inversion of each local linear model. The control output is calculated as a weighted sum of locally valid linear controllers yielding a globally nonlinear controller which results in a gain-scheduled control approach.

Hadj et al. [30] proposed the use of an artificial neural network in IMC both as process model and as controller, for a class of nonlinear systems with separable nonlinearity. It is shown that an IMC with a neural network controller, in which the linear part of the plant and its inverse are replaced by neural networks, cancels the effects of nonlinear dynamics and measured disturbances.

Denai et al. [31] developed and applied three control approaches to adjust the speed of the drive system for an induction motor. The first control design combines the variable structure theory with the fuzzy logic concept. In the second approach neural networks are used in an internal model control structure. Finally, a fuzzy state feedback controller is developed based on the pole placement technique. A simulation study of these methods is presented. The effectiveness of these controllers is demonstrated for different operating conditions of the drive system.

Baoming et al. [32] presented a nonlinear IMC for control of Switched Reluctance Motors (SRM). It is shown that combining the simplicity of the feedback linearization control and the robustness of IMC structure, gives a controller that exhibits excellent dynamic and static performances for the torque and current control. Simulations and experiments carried out on a 7.5 kW four-phase SRM, show that the ripple of the output torque is very low in spite of model-plant mismatches.

Shafiq and Riyaz [6] proposed an adaptive IMC scheme based on adaptive finite impulse response filters, which can be designed for both minimum and nonminimum phase systems in the same fashion.

Jalili-Kharaajoo et al. [33], developed an IMC scheme based on locally-linear-model-tree modelling framework. The proposed control strategy is applied to the control of PH neutralization process and the results are compared with those of IMC based on multi layer perceptron neural networks. Simulation results demonstrate the superiority of the new controller.

A neural network based multi-model IMC structure was presented by Wen et al. [34] for the adaptive tracking of plants with strong nonlinear characteristics. Multi-model is an effective method in parameters-varying and nonlinear process. The core idea is to represent a nonlinear dynamic system by a set of locally valid sub-models across the operating range.

Baoming et al. [35], achieved robust speed tracking of permanent magnet synchronous motor (PMSM) using the IMC structure.It was shown that the IMC controller greatly improves the performance of the current loop and simplifies the design procedure.

Su et al. [36], used Passivity Theorem to develop a new IMC scheme for the adaptive tracking of multivariable nonlinear processes. The conventional IMC method involves inversion of the process, which is often difficult or even impossible. In the proposed method, the process is approximated using a passive system. The controller is designed to effectively invert the passive approximation. The stability of the closed-loop system is guaranteed by the passivity condition. The effectiveness of the proposed method is illustrated by using a mixing tank control problem.

Shi and Lee [37] presented derivation of IMC controllers and tuning procedures for application to second-order plus dead-time (SOPDT) processes for achieving set-point response and disturbance rejection tradeoff.

An IMC based controller for force control in a SUMI-Ink rubbing machine was developed and implemented by Suzuki et al. [38]. It was shown that excellent force control and disturbance rejection can be achieved by the use of the IMC structure.

Jia et al. [39] presented an adaptive IMC design for hard disk drive servo control where low pass filters are used to handle high frequency mechanical modes and disturbances. They applied the scheme to the real-time tracking of the drive head.

Schwartz et al. [40] used the IMC structure for studying the effects of demand forecast error on a tactical decision policy for a single node of a manufacturing supply chain. The demand forecast is treated as an external measured disturbance in a multi-degree-of-freedom IMC based inventory control system. The multi-degree-of-freedom formulation allows the controller to be independently tuned for set-point changes, forecasted demand changes, and unforecasted demand changes. A mathematical framework for evaluating the effect of forecast revisions in an IMC controller was developed and several useful results were achieved.

Yu et al. [41] presented an adaptive internal model controller using neural networks for a tilt rotor aircraft platform. The controller includes an online learning neural network of inverse model and an off-line trained neural network of forward model. Lyapunov stability analysis was used to guarantee that tracking errors and network parameters remain bounded. The performance of the controller was demonstrated through real-time experiments.

3.4 General Nonlinear Control

Alvarez et al. [42], in 1989, proposed a tracking and regulation scheme for discrete time nonlinear systems. The scheme allows to track a specified trajectory with a dynamics specified by a tracking reference model and it was shown that the effect of disturbances on the process output can also be eliminated, with a dynamics imposed by a regulation model.

Sales and Billings [20], in 1990, introduced a minimum-variance self-tuning algorithm based on the NARMAX model. It was shown that the NARMAX based controller is more generally applicable and using NARMAX structure is a more practical approach than using functional series or block structured models. Performance analysis of the controller was discussed in terms of a cumulative loss function and high-order correlation functions of the system input, output and residual sequences.

Fruzzetti et al. [43], in 1997, proposed nonlinear model predictive controller using Hammerstein model. They used simulation studies of a pH process and a binary distillation column to illustrate the effectiveness of their controller.The motivation behind using Hammerstein model being that many chemical processes can be modelled as such, i.e. as a static non-linearity followed by a linear dynamical model. The optimization algorithm used is an ellipsoidal cutting-plane algorithm.

Banerjee and Arkun [44], in 1998, presented NMBC for control of plants that operate in several distinct operating regimes, and for the control during the transition between these regimes. The method used is to identify several linear models and interpolate the nonlinear model between these linear ones. The nonlinear model structure used is a polynomial ARX-model. The resulting NMBC controller is applied to a CSTR example.

In 2000, Ma et al. [45] presented a nonlinear self-tuning controller, which is based on Hammerstein model. A class of nonlinear systems, which can be suitably modelled with a Hammerstein model, are effectively controlled by the proposed algorithm by combining a general self-tuning method with a feedforward compensation strategy. The nonlinear parts are accommodated in the control law design so that they are compensated effectively.

Excellent reviews of the progress made in the area of nonlinear control are available in the 1998, 2001 and 2004 works of Bequette [46], Kokotovic et al. [47], and Kokotovic [48], respectively.

3.5 Neural Networks in Nonlinear Control

The first model of a neuron was devised by physiologists, McCulloch and Pitts [49] in 1943. While the "perceptron" was first developed by Rosenblatt [50]. The perceptron consists of a single artificial neuron designed to imitate, or emulate, pattern recognition tasks for biological visual systems. Rosenblatt found a simple but powerful algorithm capable of training the perceptron.

Rumelhart [51, 52] redefined Rosenblatt's perceptron replacing the hard-limiter output functions (used by perceptron) by continuous sigmoidal functions. This allowed Rumelhart to handle neural networks in an analytical manner, which inspired him to develop his back-propagation training approach. Nonlinear, multilayer neural networks could finally be trained effectively, thus making it possible to attack a wide range of problems, which only nonlinear neural networks were capable of handling properly.

It has been proved that a neural network can reproduce any nonlinear function for a limited input set. This is a direct result of the application of the universal approximation theorem [53, 54]. This theorem also predicts that a single-layer (nonlinear) neural network would be enough to produce a desired output for a given training set, though one single layer does not guarantee an optimal implementation in terms of the number of neurons and learning speed.

There are two methods for applying neural networks to control systems. They are the direct and the indirect adaptive control. Direct adaptive control can be applied when a viable model for the plant exists. Indirect adaptive control is applied when a model must be developed by a second neural network. The work for the two methods using neural networks was originally done by Narendra and Parthasarathy [55] in 1990.

There are several other researchers that have followed up the work. Tanomaru and Omatu [56] applied the methods to the inverted pendulum problem. Greene and Tan [57] applied the indirect adaptive control to a two-link-robot arm. Both methods make use back-propagation to adjust the weights of the neural networks.

Widrow and Plett [58] have presented a variety of nonlinear adaptive control schemes based on neural networks. Their main objective was to control the plant dynamics and the internal disturbance, without compromising either of the two.

A detailed study of the application of neural networks to nonlinear control can be found in [59].

CHAPTER 4

U-MODEL BASED ADAPTIVE IMC USING RADIAL BASES FUNCTIONS

Internal Model Control (IMC) is one of the popular control strategies used in industrial process control. Its main features are its simple structure, fine disturbance rejection capabilities and robustness with respect to parameter variations [8-13]. IMC can be used for both linear and nonlinear systems [14], and is especially suitable for design and implementation of controller for open-loop stable systems. As reviewed in section 3.3, the IMC has been used in a number of situations that include those with known plants and those with uncertain plant dynamics. Our interest here is in developing an adaptive version of the IMC that makes use of the control-oriented nature of the U-Model. Such a structure, if combined with a good identification scheme, will provide an excellent design platform of controllers for adaptive tracking of nonlinear dynamic plants. In this chapter we first propose such a scheme and then provide details of its various components. We also demonstrate the performance of the proposed scheme with the help of simulations and real-time applications.

4.1 Proposed Adaptive IMC Structure

The proposed structure is depicted in figure 4.1. This is essentially an adaptive version of the IMC where the unknown plant $f_C(.)$ is modelled using nonlinear radial bases functions as $f_M(.)$. A U-Model equivalence of the model $f_M(.)$ is then used to convert

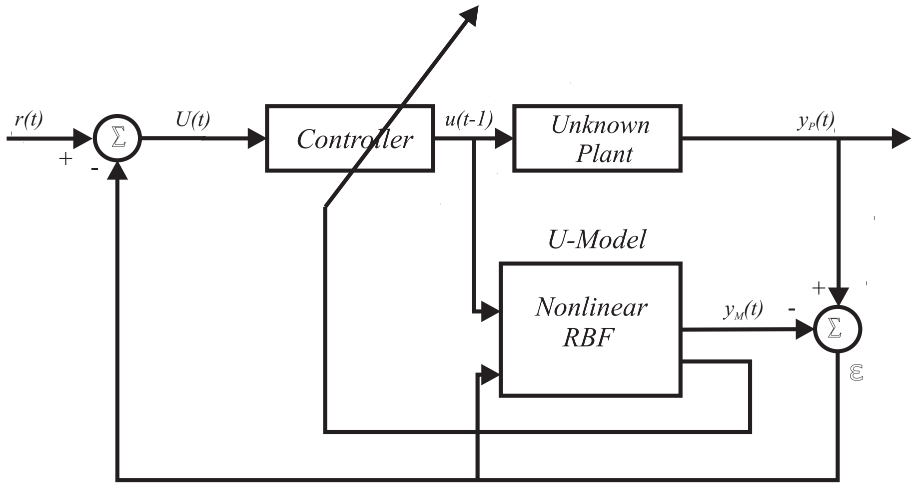

Figure 4.1: Proposed Structure Block Diagram

the controller synthesis part into a simple root-solving procedure. A standard leaky normalized LMS is used as the adaptation algorithm for the update of the model $f_M(.)$ parameters.

The components of the proposed scheme are now discussed in detail.

4.1.1 The Overall Structure: IMC

Since the proposed scheme is based on the IMC strategy we discuss the operation and properties of the IMC. The basic IMC structure is depicted in figure 4.2. Here $r(t)$ is the reference signal and $d(t)$ represents external noise or disturbance.

4.1.1.1 Asymptotic Tracking

If the model were the exact representation of the plant (i.e. $f_M(.) = f_P(.)$) and the controller is the inverse of the model (i.e. $f_C(.) = [f_M(.)]^{-1}$), then the transfer function from the input $U(t)$ to the plant output $y(t)$ can be considered as a simple delay q^{-L} (considering no external noise $d(t)$, for simplicity). However, since there

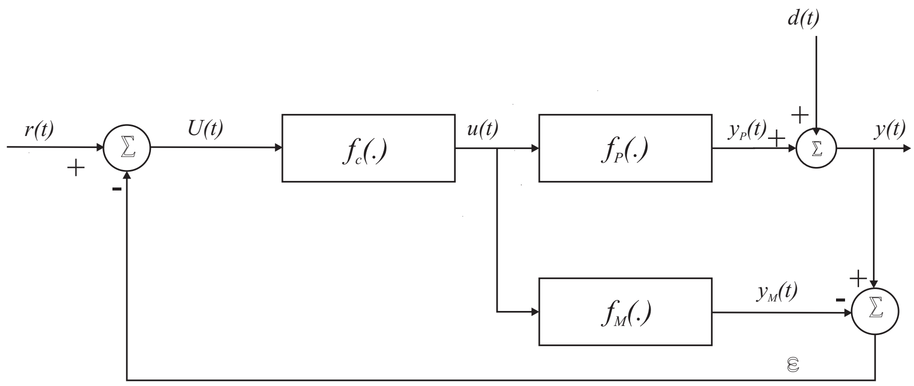

Figure 4.2: Internal Model Control

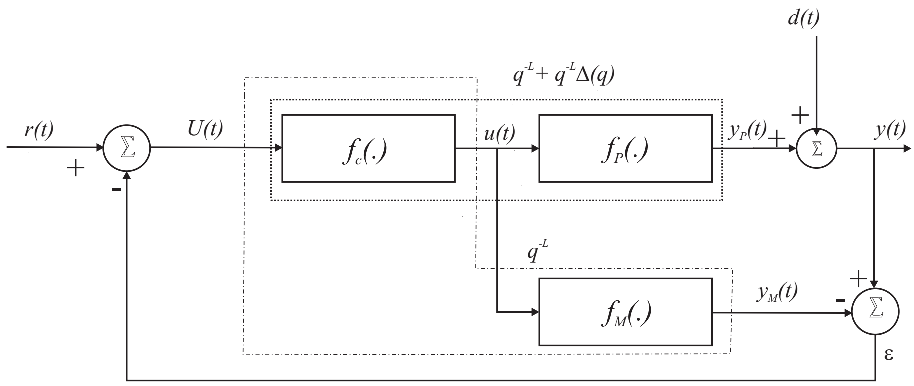

Figure 4.3: IMC: Tracking

can be uncertainty in the plant representation, the actual transfer function from the $U(t)$ to $y(t)$ will also include $\Delta(q)$ (the plant uncertainty). This is depicted in figure 4.3 and is given mathematically as:

$$y(t) = [q^{-L} + q^{-L}\Delta(q)]U(t) \tag{4.1}$$

$$y_M(t) = q^{-L}U(t) \tag{4.2}$$

substituting equations (4.1) and (4.2) in $\epsilon(t) = y(t) - y_M(t)$ gives,

$$\epsilon(t) = q^{-L}\Delta(q)U(t) \tag{4.3}$$

Using the above given equations, the overall closed loop gain can be written (for $L = 1$) as:

$$y(t) = r(t-1)[1+\Delta(q)] - r(t-2)[\Delta(q)+\Delta^2(q)][\frac{1}{1+q^{-1}\Delta(q)}] \tag{4.4}$$

Here, if $|\Delta(q)| \ll 1$ then we have $y(t) \approx r(t-1)$ (note that this requires a very good identification scheme). Thus we have shown that approximate tracking can be achieved using the IMC strategy.

4.1.1.2 Summary of IMC Properties [12]

Defining the controller, plant and model gains by C,P and M respectively the following relations can be obtained readily from figure 4.2

$$U = r - y + y_m \tag{4.5}$$

$$y_m = M\left(C\left(U\right)\right) \tag{4.6}$$

$$\epsilon = \left(P - M\right)\left(C\left(U\right)\right) + d \tag{4.7}$$

These lead to the following properties of the IMC structure,

1. *Property P1 (Dual Stability):* If the plant and the controller are input-output stable and the model is a perfect representation of the plant; then the closed-loop system is input-output stable (because then the system becomes open-loop)

2. *Property P2 (Perfect Control):* If the inverse of the operator describing the plant model exists, and this inverse is used as the controller, and the closed-loop system is input-output stable with this controller; then the control will be perfect (set $C = M^{-1}$ in (4.6) and use the result in (4.5)).

3. *Property P3 (Zero Offset):* If the inverse of the steady-state model operator exists, and the steady-state controller operator is equal to this inverse, and the closed-loop system is input-output stable with this controller. Then offset free control is attained for asymptotically constant inputs.

4.1.2 The Plant

The IMC strategy does not include a stabilizing mechanism and is designed essentially for open-loop stable plants (which is the case for a large number of industrial systems). Thus the proposed scheme is applicable to stable nonlinear dynamic plants. It is also applicable if the given plant happens to be unstable, and it can be stabilized using some known stabilization technique. This stabilized plant could be used in the proposed IMC and the tracking objective will be achieved.

4.1.3 RBF Based Nonlinear Model

A proper identification of the uncertain plant plays a critical role in any adaptive control technique. The main idea involved in identification is the following: given a set of inputs and desired outputs, we must find a mapping $\hat{F}$ between them such that a measure of the error between the resulting output and the actual output is minimal. A large number of function approximation methods are available in the literature. In particular, the mapping can be achieved through nonlinear functions used in neural networks, such as the radial bases functions.

A radial basis function $\varphi(\|x - m_i\|)$ has the same value for all neural inputs x that lie on a hypersphere with center m_i, i.e., it exhibits radial symmetry. A nonlinear model based on radial bases functions can take the form depicted in figure 4.4. Here the input variables are preprocessed through the nonlinear Gaussian radial

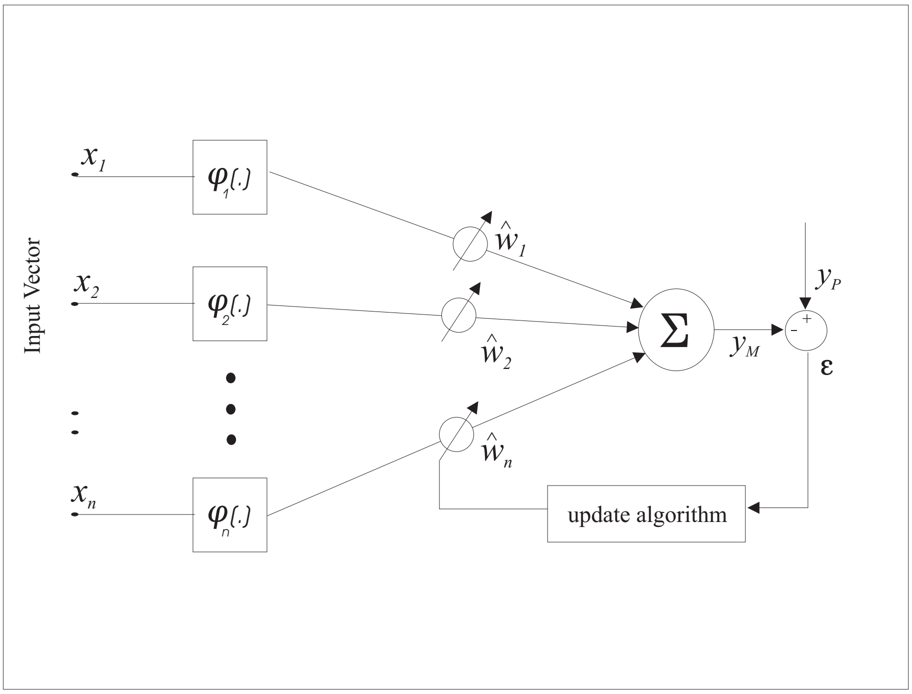

Figure 4.4: RBF Based Nonlinear Filter

basis functions. These are then input to the adaptive linear combiner that adaptively assigns weights to these inputs. The resulting sum gives the approximated output which can be written as:

$$\hat{F}(x) = \sum_{i=1}^{n} \hat{w}_i \varphi \left(\|x - m_i\| / \sigma_i \right) \tag{4.8}$$

with

$$\varphi(x) = \exp\left(\frac{\|x - \mu\|^2}{\sigma^2} \right) \tag{4.9}$$

The weights can be updated iteratively using a suitable form of the Least Mean Square (LMS) algorithm. This algorithm seeks to minimize the mean squared error $E\{|y - \hat{F}(x)|^2\}$ (a detailed discussion follows in section 4.2).

In the light of the preceding discussion, it is proposed to perform an online identification of the plant using an RBF based nonlinear model. The RBF model is given by

$$y_M(t) = a_1 u(t-1) + \hat{b}_2 \varphi(u(t-2)) + \hat{b}_3 \varphi(u(t-3)) + \ldots + \hat{b}_n \varphi(u(t-n)) \quad (4.10)$$

where the parameter a_1 is selected in advance and the parameters $\hat{b_2}, \hat{b_3}, ..., \hat{b_n}$ are estimated using some suitable form of the LMS algorithm (this is discussed in detail in section 4.2). $\varphi(.)$ can be any function used in neural networks. Here the use of Gaussian RBF (4.9) is proposed.

4.1.4 U-Model Based Controller

Having selected a proper identification model, the next important step is to synthesize a suitable control law for the IMC controller $f_C(.)$. As pointed out in chapter 1, one of the more important considerations in adaptive control design is the ability to define a general yet simplistic control law for the system. To meet this objective, we propose to make use of the control-oriented nature of the U-Model. Therefore, in this section we first develop a U-Model equivalence for equation (4.10) and then use this equivalence to synthesize a simplistic controller.

4.1.4.1 U-Model Equivalence of the Proposed Model

To simplify the synthesis of the control law, we introduce the equivalent U-model for the radial basis nonlinear model (4.10) as:

$$y_M(t) = \alpha_0(t) + \alpha_1(t) u(t-1) \quad (4.11)$$

where

$$\begin{aligned}\alpha_0(t) =& \hat{b}_2\varphi(u(t-2)) + \hat{b}_3\varphi(u(t-3)) \\ & + \ldots + \hat{b}_n\varphi(u(t-n)) \\ \alpha_1(t) =& a_1\end{aligned}$$

4.1.4.2 Controller Synthesis

The controller output $u(t-1)$ can be easily obtained from (4.11) as:

$$u(t-1) = \frac{y_M(t) - \alpha_0(t)}{\alpha_1(t)} \tag{4.12}$$

Now referring to the IMC block diagram (figure 4.2), we want to have $f_C(.) = [f_M(.)]^{-1}$. This can be assured by writing the control signal in terms of $U(t)$ as:

$$u(t-1) = \frac{U(t) - \alpha_0(t)}{\alpha_1(t)} \tag{4.13}$$

Eq. (4.13) gives the control law needed for the proposed scheme.

4.2 Selection of Adaptation Algorithm

The selection of a proper adaptation algorithm for the proposed scheme is critical to its overall performance. We must incorporate an adaptation scheme that is stable, provides good convergence behavior and lends itself to easy implementation. Since the proposed model is essentially in the form of a nonlinear combiner, we can make use of the rich literature in the area of adaptive filtering [60–62].

In this section, we develop a Wiener solution for the weights and then introduce Least Mean Square (LMS) solution to the problem. This is followed by inclusion of two improvements to the LMS algorithm, namely, *normalization* and *leakage*. In order to have notational consistency with the bulk of literature available, we shall make use of the following standard notations for our system.

The input vector $[\varphi(u(t-2)) \quad \varphi(u(t-3)) \ldots \varphi(u(t-n))]^T$ is denoted by $\mathbf{x}(k) = [x_0(k) \quad x_1(k) \ldots x_n(k)]^T$ and the weight vector $[\hat{b}_2 \quad \hat{b}_3 \ldots \hat{b}_n]^T$ is denoted by $\mathbf{w}(k) = [w_0(k) \quad w_1(k) \ldots w_n(k)]^T$.

4.2.1 The Wiener Solution

Given a set of desired outputs $d(k)$ and a linear combiner model:

$$y(k) = \sum (w_i(k)x_i(k)) = \mathbf{w}^T(k)\mathbf{x}(k) \tag{4.14}$$

the Wiener solution finds the optimal weight vector $\mathbf{w}_o$ that minimizes the mean square error (MSE). MSE is one of the most widely used objective functions in adaptive filtering and is defined as:

$$\xi(k) = E[e^2(k)] = E[d^2(k) - 2d(k)y(k) + y^2(k)] \tag{4.15}$$

Using (4.14), equation (4.15) can be rewritten as:

$$\begin{aligned} \xi(k) =& E[d^2(k) - 2d(k)\mathbf{w}^T(k)\mathbf{x}(k) + \mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k)] \\ =& E[d^2(k)] - 2E[d(k)\mathbf{w}^T(k)\mathbf{x}(k)] + E[\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k)] \end{aligned} \tag{4.16}$$

for a filter with fixed coefficients, the MSE function can be given as:

$$\begin{aligned} \xi =& E[d^2(k)] - 2\mathbf{w}^T E[d(k)\mathbf{x}(k)] + \mathbf{w}^T E[\mathbf{x}(k)\mathbf{x}^T(k)]\mathbf{w} \\ =& E[d^2(k)] - 2\mathbf{w}^T\mathbf{p} + \mathbf{w}^T\mathbf{R}\mathbf{w} \end{aligned} \tag{4.17}$$

where $\mathbf{p} = E[d(k)\mathbf{x}(k)]$ is the cross-correlation vector between the desired and the input signal, and $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^T(k)]$ is the input signal correlation matrix. It may be noted the the objective function ξ is a quadratic function of the weight coefficients. This should allow a straightforward solution if vector $\mathbf{p}$ and matrix $\mathbf{R}$ are known. To find the optimal weight vector that gives the minimal MSE we take the gradient of the objective function ξ with respect to the weight vector $\mathbf{w}$.

$$\mathbf{g_w} \triangleq \frac{\partial \xi}{\partial \mathbf{w}} = \left[\frac{\partial \xi}{\partial w_0} \quad \frac{\partial \xi}{\partial w_1} \cdots \frac{\partial \xi}{\partial w_n} \right]^T = -2\mathbf{p} + 2\mathbf{Rw} \tag{4.18}$$

By setting the gradient vector equal to zero and assuming that $\mathbf{R}$ is nonsingular, the desired optimal weight vector is obtained as:

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p} \tag{4.19}$$

The solution above is known as the Wiener solution. This solution depends upon the values of $\mathbf{R}$ and $\mathbf{p}$. However, in practice, precise estimations of these two are not available. This leads to the LMS algorithm which we discuss next.

4.2.2 The LMS Algorithm

If good estimates of matrix $\mathbf{R}$, denoted by $\hat{\mathbf{R}}(k)$, and of vector $\mathbf{p}$, denoted by $\hat{\mathbf{p}}(k)$, are available, a steepest-descent algorithm can be used to *search* the Wiener solution of equation (4.19) as follows [61]:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \mathbf{g_w}(k) = \mathbf{w}(k) + 2\mu(\hat{\mathbf{p}}(k) - \hat{\mathbf{R}}(k)\mathbf{w}(k)) \tag{4.20}$$

$$k = 0, 1, 2, \ldots,$$

where $\mathbf{g_w}(k)$ represents an estimate of the gradient vector of the objective function with respect to the filter coefficients. Mostly, the following instantaneous estimates for $\mathbf{R}$ and $\mathbf{p}$ are used to estimate the gradient vector:

$$R(k) = x(k)x^T(k);\ p(k) = d(k)x(k) \tag{4.21}$$

Using these estimates in equation (4.20), we get the update equation based on the LMS algorithm,

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\mu(k)\mathbf{x}(k)e(k) \tag{4.22}$$

where $e(k) = x^T(k)w(k) - d(k)$ and the convergence factor μ should be chosen in a range to guarantee convergence. This range is given by [61, 63]:

$$0 < \mu < \frac{1}{tr[\mathbf{R}]} = \frac{1}{1 + \sum_{i=1}^{n} E[x_i^2(k)]} \tag{4.23}$$

4.2.3 Ensuring Convergence of the LMS through Persistence of Excitation

With the help of LMS algorithm we have in our hands an iterative solution for the optimal weight vector. However, before we can use these results, the following points need to be considered:

1. The optimal Wiener solution requires the input autocorrelation matrix **R** to be nonsingular.

2. Good convergence behavior of the LMS depends upon the proper selection of μ and on the well-behavidness of matrix **R**.

A natural question arises that in the proposed scheme, how do we actually ensure the above mentioned requirements? One approach could be to study the nature of inputs being sent to the adaptive filter and see if the expected inputs produce well-behaved autocorrelation matrix. This will involve, for the case of the proposed adaptive IMC, an analysis of the following matrix:

$$\mathbf{R} = E\begin{bmatrix} 1 & \varphi(u(t-2)) & \cdots & \varphi(u(t-n)) \\ \varphi(u(t-2)) & \varphi(u(t-2))\varphi(u(t-2)) & \cdots & \varphi(u(t-2))\varphi(u(t-n)) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(u(t-n)) & \varphi(u(t-n))\varphi(u(t-2)) & \cdots & \varphi(u(t-n))\varphi(u(t-n)) \end{bmatrix}$$

However, an analytically simpler and more effective approach can be presented if we make use of the concept of persistence of excitation (PE) that appears so often in system identification theory. For this, first of all we note that the convergence of the LMS algorithm (and the well-behavedness of matrix $\mathbf{R}$) is in fact linked with the persistence of excitation of the input. It is well-known that white noise has the highest PE because it can excite all the modes of the system. Thus if we could introduce random noise into the system, it will add PE to the input and will ensure convergence of the LMS algorithm. Two commonly used approaches for this purpose are [60, 62]:

1. Addition of a controlled random noise (called dither) to the input.

2. Introduction of weight leakage in the LMS update equations, which has the same effect as in the dithering approach.

We propose to use the weight leakage approach to ensure the convergence of the LMS. This method is commonly known as the Leaky LMS (LLMS) algorithm and is discussed next.

4.2.4 Leaky LMS Algorithm: LLMS

The LLMS algorithm is a variant of the LMS algorithm and is obtained by redefining the objective function as (with α as a small positive constant) :

$$\xi = E[e^2] + \alpha E[\mathbf{w}^T \mathbf{w}] \tag{4.24}$$

In other words, the objective function now includes a new term proportional to the norm of the weight vector. The optimal Wiener solution in this case is obtained as done earlier by taking the gradient ξ with respect to $\mathbf{w}$.

$$\frac{\partial \xi}{\partial \mathbf{w}} = -2\mathbf{p} + 2\mathbf{R}\mathbf{w} + 2\alpha\mathbf{w} = -2\mathbf{p} + 2(\mathbf{R} + \alpha\mathbf{I})\mathbf{w} \tag{4.25}$$

Equating this gradient to zero gives:

$$\mathbf{w}_o = (\mathbf{R} + \alpha\mathbf{I})^{-1}\mathbf{p} \tag{4.26}$$

Observing equation (4.26), the application of the leaky LMS algorithm results in the addition of small constant α to the terms on the main diagonal of the correlation matrix of the input process; one obtains the same result by summing white noise with statistical power α to the input process. Thus the approach is useful in ensuring a well-behaved matrix $\mathbf{R}$ and accelerates the convergence of the LMS algorithm [62]. The update equations for the leaky LMS can be easily obtained as:

$$\mathbf{w}(k+1) = (1 - \mu\alpha)\mathbf{w}(k) + \mu\mathbf{x}(k)e(k) \tag{4.27}$$

It must be noted that the LLMS adds certain degradation to the LMS algorithm and is slightly more difficult in implementation. However, it provides better overall performance in limited precision situations and is, therefore, desirable.

4.2.5 Improving LLMS Algorithm Convergence through Normalization

The adaptation algorithm to be used has been developed so far as the LLMS with update equation:

$$\mathbf{w}(k+1) = (1-\mu\alpha)\mathbf{w}(k) + \mu\mathbf{x}(k)e(k)$$

Since fast model convergence is an important part of online identification schemes, we propose to use normalization with the update equation above. The idea behind normalization is to use a variable convergence factor (say μ_k) in place of the constant convergence factor μ. The variable convergence factor is then chosen to accelerate the convergence rate. This has been shown in literature, to improve the convergence of the LMS significantly at the cost of very small degradation in the final results [61]. The final form of the algorithm we propose to use with the adaptive IMC scheme (generally known as the normalized Leaky-LMS: nLLMS) is given by:

$$\mathbf{w}(k+1) = (1-\mu\alpha)\mathbf{w}(k) + \mu_k\mathbf{x}(k)e(k) \tag{4.28}$$

where

$$\mu_k = \frac{\mu}{\gamma + \mathbf{x}^T\mathbf{x}} \tag{4.29}$$

and μ is a small positive constant typically selected in the range $0 < \mu < 2$, to ensure convergence in the mean square sense. A very small positive constant γ is added to the denominator to overcome potential numerical instability in the weight update (e.g. in Matlab, $\gamma = 1 \times 10^{-10}$).

4.3 Discussion on the Stability of the Proposed Scheme

Stability analysis forms an important part of any control scheme. In this section, we present discussions on the stability of proposed adaptive scheme. Since the system is nonlinear, specific stability test tools have to be utilized. In the following, our discussion is based on two important approaches to the study of stability of nonlinear adaptive systems. These are:

1. The Small Gain Theorem: The small gain theorem is a very useful criterion in studying stability of systems. Its main feature is that it applies to all feedback systems; linear as well as nonlinear. It states that a general feedback system is input-output stable if the open-loop gain is less than unity. The first analysis of the nonlinear IMC performed by [12], used this criterion to propose measures towards ensuring stability.

2. Internal Stability: This criterion often appears in modern literature on the stability of adaptive schemes such as the IMC (e.g., [28, 41]). An internally stable system is defined as the system in which bounded signals injected at any point of the control system will generate bounded responses at any other point [64]. By applying L-space theory [65], the internally stable system means that all signals at any point of control system should be bounded.

Applying the small gain theorem to the IMC in figure 4.2, we have [12]:

$$g((F_P - F_M)F_C) < 1 \tag{4.30}$$

where $g(H)$ represents the gain of the nonlinear operator H (for a precise definition of the gain of a nonlinear system see [10]). This leads to the sufficient condition:

$$g(F_C)g(F_P - F_M) < 1 \tag{4.31}$$

From here the first property of the IMC (discussed in section 4.1.1.2) becomes clear, i.e., when $F_P = F_M$, the system is stable. However, when the model-plant mismatch is large, in the sense of $g(F_P - F_M)$ being large, the controller gain has to be small and this may not always be ensured. To overcome this potential instability, the most common approach in the literature is to use a tunable robustness filter F_R (usually a first-order low pass filter) just before the controller. This leads to the sufficient condition:

$$g(F_R)g(F_C)g(F_P - F_M) < 1 \tag{4.32}$$

Thus, as long as the controller and $F_P - F_M$ are stable, we can always find a robustness filter satisfying the inequality.

The treatment above does not make use of the fact that the adaptation we have used (the nLLMS) is stable and converges in the mean square sense. The convergence of the nLLMS was discussed in section 4.2.3 while a detailed discussion on the stability of the nLLMS filter can be found in [66]. We can make use of this fact in the internal stability test to demonstrate the stability of the proposed scheme. This is done in the following steps

1. The stability of the nLLMS based filter (see [66]) ensures that the error term converges in the mean square sense, i.e., $\varepsilon(t) \epsilon L_2$.

2. Since the controller is input-output stable (this follows from equation (4.13) and the stability of the nLLMS based model), this, coupled with the fact that

$\varepsilon(t) \epsilon L_2$ and $r(t) \epsilon L_\infty$, leads to a bounded controller output $u(t) \epsilon L_\infty$.

3. When this bounded input $u(t) \epsilon L_\infty$ enters the input-output stable plant, it gives the bounded output $y_P(t) \epsilon L_\infty$.

Thus we have shown that bounded inputs injected at any point in the system give rise to bounded outputs and that the system is input-output stable.

4.4 Application of the Proposed Scheme

In order to demonstrate the effectiveness of the proposed scheme, we carried out simulations as well as real-time experiments. The nonlinear plants used and the results in each case are now discussed in detail.

4.4.1 Adaptive Tracking of the Hammerstein Model

The class of Hammerstein models was presented in chapter 2. Its main importance stems from the fact that a larger number of nonlinear processed can be modelled as a nonlinear part followed by a linear part. A case of the Hammerstein model used for this study is given as:

$$y(t) = 0.5y(t-1) + x(t-1) + 0.1x(t-2)$$
$$x(t) = 1 + u(t) - u^2(t) + 0.2u^3(t)$$

The system is modelled according to (4.10) as:

$$\begin{aligned} y_M(t) =& a_1 u(t-1) + \hat{b}_2\varphi(u(t-2)) + \hat{b}_3\varphi(u(t-3)) + \hat{b}_4\varphi(u(t-4)) \\ &+ \hat{b}_5\varphi(u(t-5)) \end{aligned}$$

The functions $\varphi(.)$ are selected as the Gaussian RBF with zero mean and unit variance. The U-Model equivalence of the model above can be given as:

$$y_M(t) = \alpha_0(t) + \alpha_1(t)u(t-1)$$

with

$$\alpha_0(t) = \hat{b}_2\varphi(u(t-2)) + \hat{b}_3\varphi(u(t-3)) + \hat{b}_4\varphi(u(t-4)) + \hat{b}_5\varphi(u(t-5))$$

and

$$\alpha_1(t) = a_1$$

With this equivalence, the controller of equation (4.13) can be used. The first parameter a_1 is selected as 5. All weights are initialized to 0 and the step size is chosen to be 0.1. The leakage factor is 0.99 and an arbitrary input signal is used. Adaptive tracking is achieved and the results are given in figures 4.5, 4.6, and 4.7.

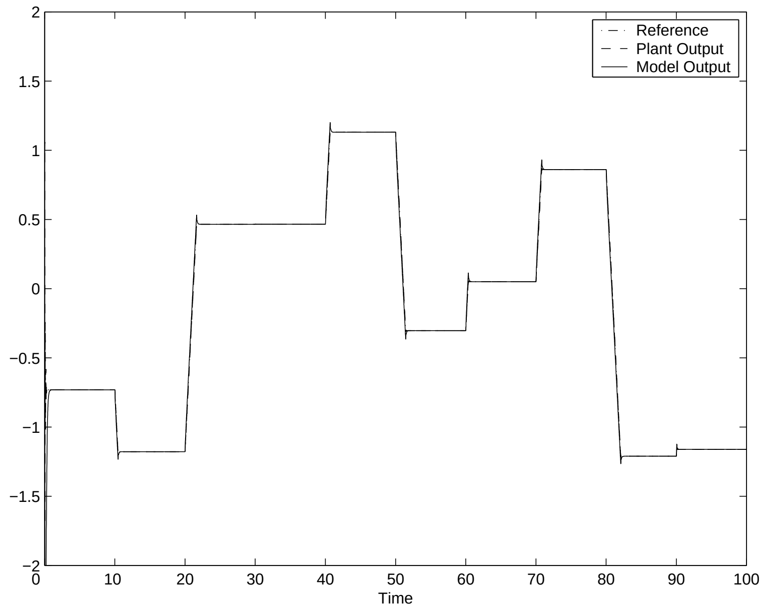

Figure 4.5: Hammerstein Model: Tracking

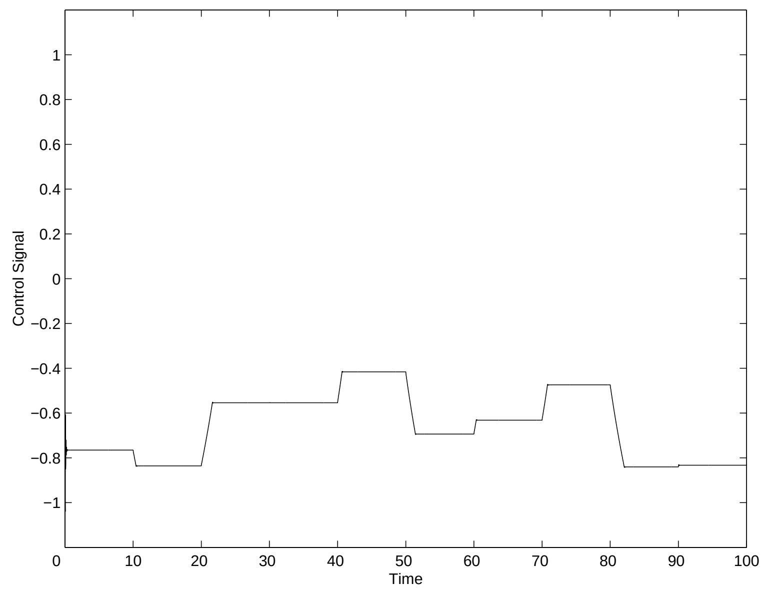

Figure 4.6: Hammerstein Model: Control Input

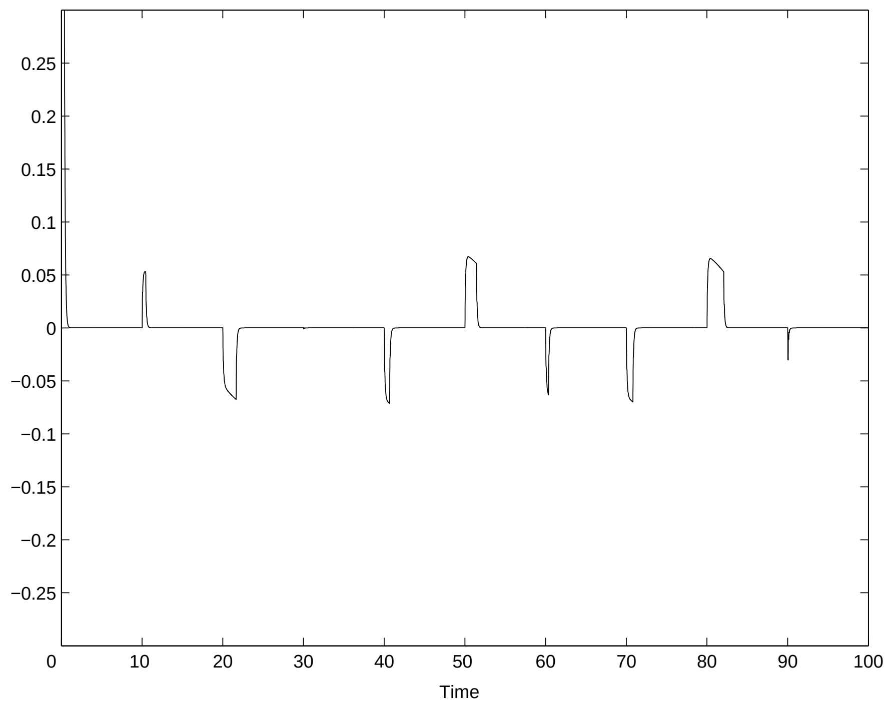

Figure 4.7: Hammerstein Model: Plant-Model Output Mismatch

4.4.2 Adaptive Tracking of DC Motor Speed and Position

The need for a simplistic control structure for nonlinear plants becomes clearer when an effort is made to precision control a dynamic nonlinear plant exhibiting high uncertainties, such as the DC motor. Today high performance electric motor drives find a wide scope of applications in areas including mechatronics (robotics, rolling mills etc.) and manufacturing (high precision machining, small component assembly etc.). All these applications demand an accurate control of speed and/or position. However, the motor control problem is characterized by variable and unpredictable inputs, noise propagation along a series of unit processes, unknown parameters, and changes in motor/load dynamics. Under these conditions, the conventional constant gain feedback controller fails to maintain the performance of the system at acceptable levels and does not respond well to changes in system dynamics. This has resulted in an increased interest in developing *adaptive* control systems for motor drives [67–73]. However, most of the existing adaptive schemes either use linear models or complicated nonlinear models such as the NARMAX. The linear model approach is not suitable for high precision applications as it unrealistically neglects the nonlinear dynamics. On the other hand controller design based on generalized nonlinear models such as the NARMAX is not an easy task. Therefore, we apply the proposed adaptive scheme to the speed and position tracking of the DC motor.

4.4.2.1 DC Motor Speed Tracking: Simulations

The discrete time model [73] used for the simulations is given by:

$$\begin{aligned}\omega_r(n+1) = {} & K_1\omega_r(n) + K_2\omega_r(n-1) + K_3\left[sign\left\{\omega_r\left(n\right)\right\}\right]\omega_r^2(n) \\ & + K_4\left[sign\left\{\omega_r\left(n-1\right)\right\}\right]\omega_r^2(n-1) + K_5 v_a\left(n\right) \qquad (4.33)\end{aligned}$$

where $v_a(t)$ is the time-varying motor terminal voltage and $\omega_r(t)$ is the motor speed. K_1, K_2, K_3, K_4 and K_5 are constants expressed in terms of motor parameters and are given as (T_s is the sampling period):

$$
\begin{aligned}
K_1 &= \frac{2(L_aJ + R_aJ + L_aD - R_aDT_s + K^2T_s)}{L_aJ + 2R_aJ + 2L_aD)} \\
K_2 &= -\frac{L_aJ}{L_aJ + 2R_aJ + 2L_aD)} \\
K_3 &= -\frac{2\mu(L_a + R_aT_s)}{L_aJ + 2R_aJ + 2L_aD)} \\
K_4 &= \frac{2L_a\mu}{L_aJ + 2R_aJ + 2L_aD)} \\
K_5 &= \frac{2KT_s}{L_aJ + 2R_aJ + 2L_aD)}
\end{aligned}
$$

where

$R_a =$ armature resistance in $ohms$

$D =$ viscous constant in $N.m.s$

$K =$ torque constant in $N.m/A$

$\mu =$ load torque constant in $N.m.s^2$

$L_a =$ armature inductance in H

$J =$ rotor inertia in $kg.m^2$

The system is modelled according to (4.10), and then its equivalent U-Model (4.11) is used to synthesize the control law (4.13). For simulation purpose a sampling time of 0.01 seconds is selected. The first parameter a_1 is set to 1.055, while the number of linear combination weights is selected to be four $(\hat{b}_2, \hat{b}_3, \hat{b}_4, \hat{b}_5)$. All weights are initialized randomly (between 0 and 1) and the step size for the nLMS is chosen to be 0.05. The leakage factor is set to 0.99. Adaptive tracking was achieved and the results are shown in figures 4.8, 4.9 and 4.10. Figure 4.8 shows the reference signal, the DC motor output and the model output. Figure 4.9 shows the mismatch between the plant and model outputs while figure 4.10 shows the controller signal (in volts).

4.4.2.2 DC Motor Position Tracking: Simulations

The position tracking of DC Motors is in general more difficult then the speed tracking this is because the system is originally unstable. The proposed scheme requires the plant to be stable and for this purpose the plant is first stabilized using simple feedback.

For simulation purposes the model (4.33) can be used with the introduction of the position parameter θ (taken in radians). It is related to the speed as $\omega_r = \dot{\theta}$ in continuous time, while for discrete time, divided difference approach can be used to get (with T_s as the sampling period)

$$\omega_r(n) = \frac{\theta(n) - \theta(n-1)}{T_s}$$

This time again, four weights are used (initialized randomly between 0 and 1), and the step size is 0.18. The leakage factor is set to 0.97 and $a_1 = 1.5$ is found suitable for this application. Resulting plots are given in figures 4.11, 4.12 and 4.13.

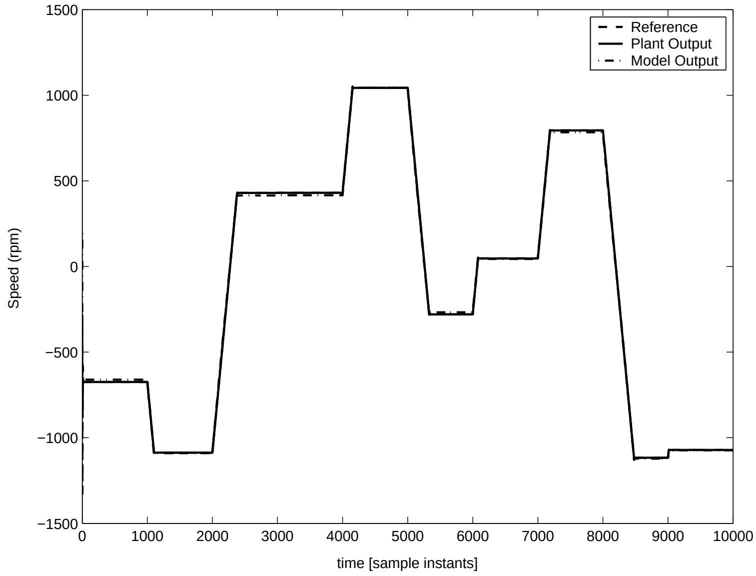

Figure 4.8: DC Motor Speed: Tracking

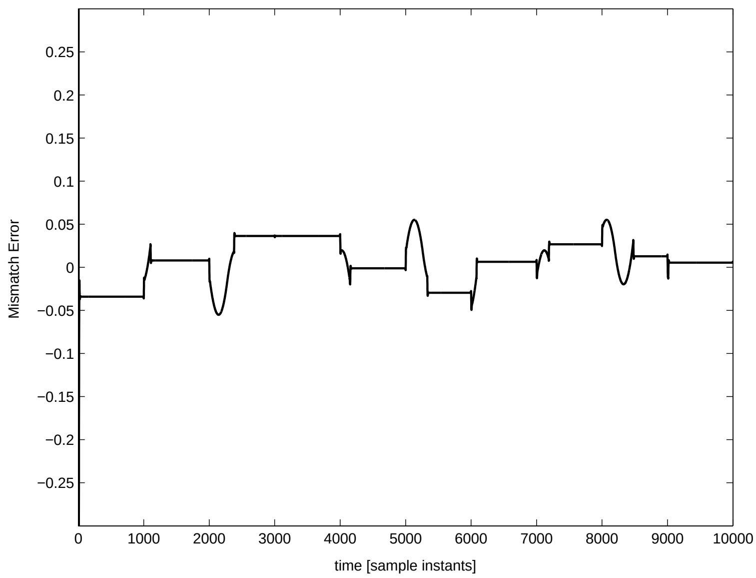

Figure 4.9: DC Motor Speed: Plant-Model Output Mismatch

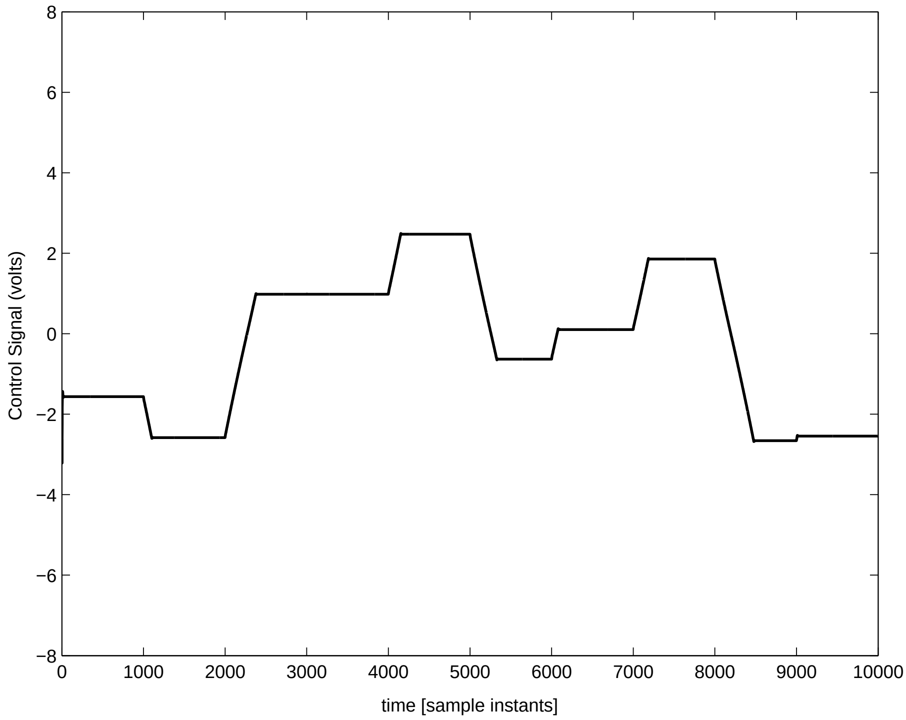

Figure 4.10: DC Motor Speed: Control Signal

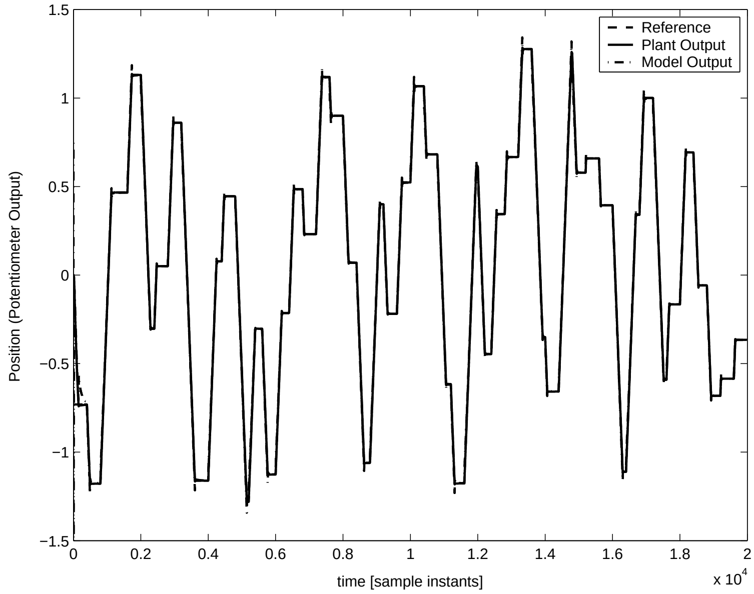

Figure 4.11: DC Motor Position: Tracking

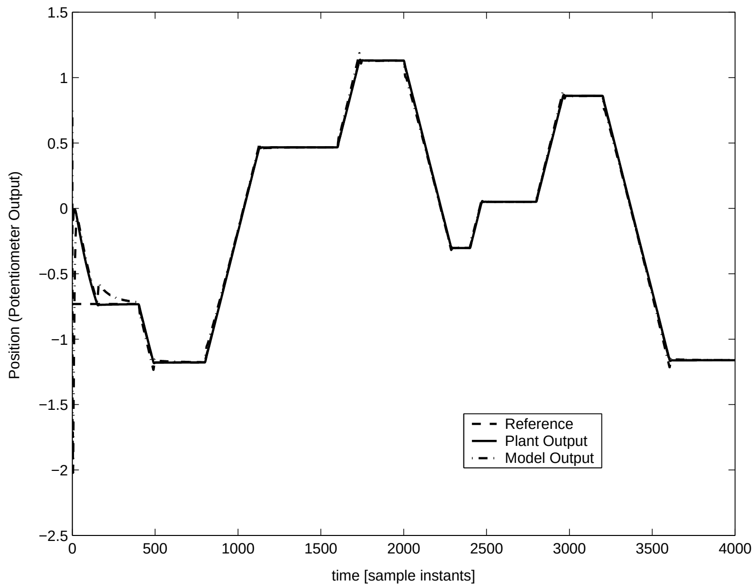

Figure 4.12: DC Motor Position: Magnified View of Tracking

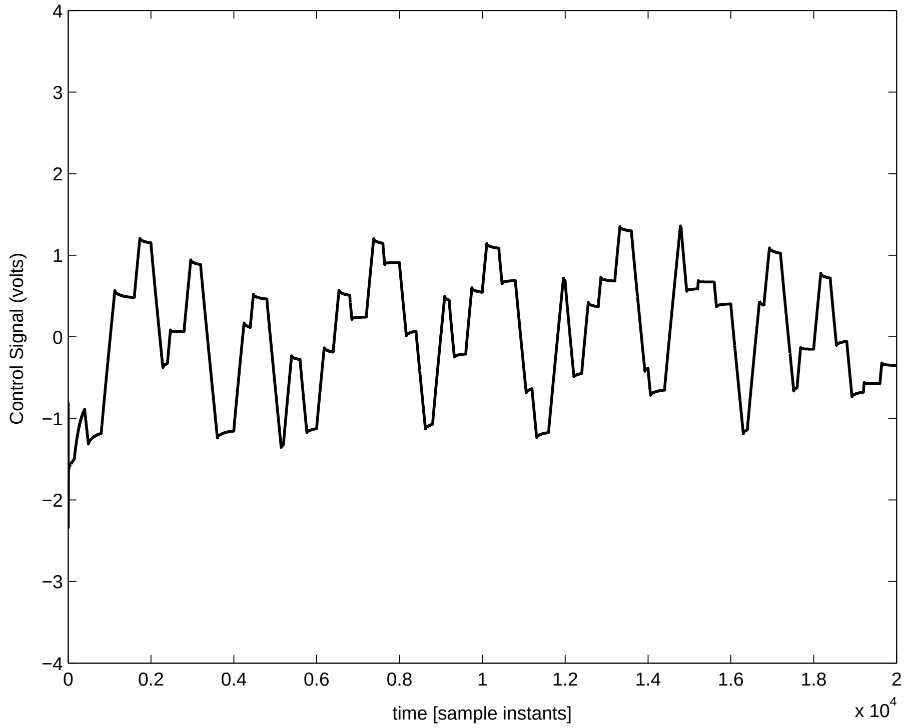

Figure 4.13: DC Motor Position: Control Signal

4.4.3 Real-Time Adaptive Tracking of the DC Motor Speed Under Fixed and Variable Load Conditions

The scheme is also applied in real-time to the adaptive speed control of the Feedback-DCM150F (Crouzet 8285002) brush DC motor. Standard IBM PC-type Pentium III is used for real time computation, while data acquisition is accomplished through Advantech card PCI-1711. Simulink's real-time window target environment is used to implement the controller. Motor speed is measured by a tachometer (which generates voltage proportional to motor speed at $2.5V/1000rpm$) and a servo amplifier is used to provide variable voltage (control input) for the excitation of motor. A sampling period of 0.01 seconds is selected. The first parameter a_1 was set to 10, while the number of linear combination weights is four $(\hat{b_2}, \hat{b_3}, \hat{b_4}, \hat{b_5})$. All weights were initialized randomly (between 0 and 1) and the step size for the nLLMS is chosen to be 0.8 with a leakage factor of 0.99. The experiment results are given in figures 4.14, 4.17, 4.16 and 4.15. Here, figure 4.14 gives the reference signal, the DC motor output and the model output (converted to rpm, actual signals are in volts). figure 4.17 shows the mismatch error between the plant and the model outputs, while figure 4.16 gives the control input (both in volts). Finally, figure 4.16 is just a magnified view of the initial part of figure 4.14 and is meant to show the details of adaptation.

To further verify the robustness of the proposed scheme to dynamic load variations, a magnetic brake was used to decelerate the metallic disc being rotated by the DC motor. Due to the speed dependence of the magnetic brake force, the motor experiences different loading at different reference speeds (time varying parameter). The system was able to provide good tracking and the results are shown in figures 4.18 and 4.19.

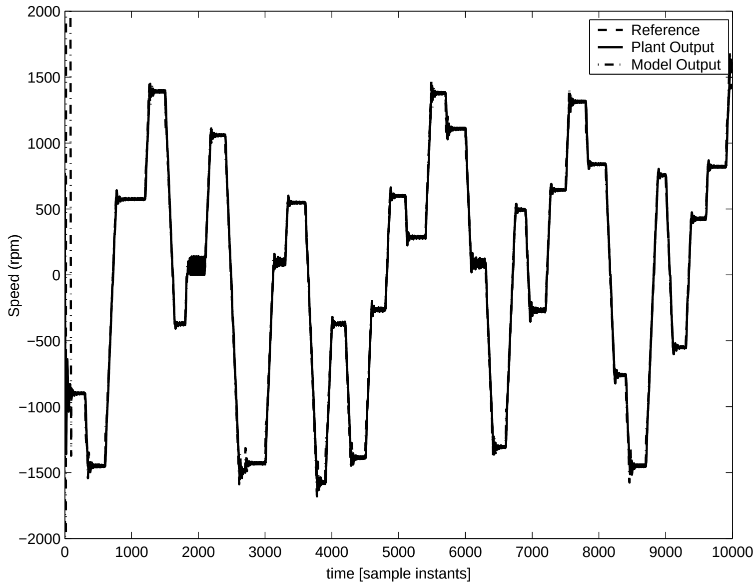

Figure 4.14: Fixed Load: Tracking

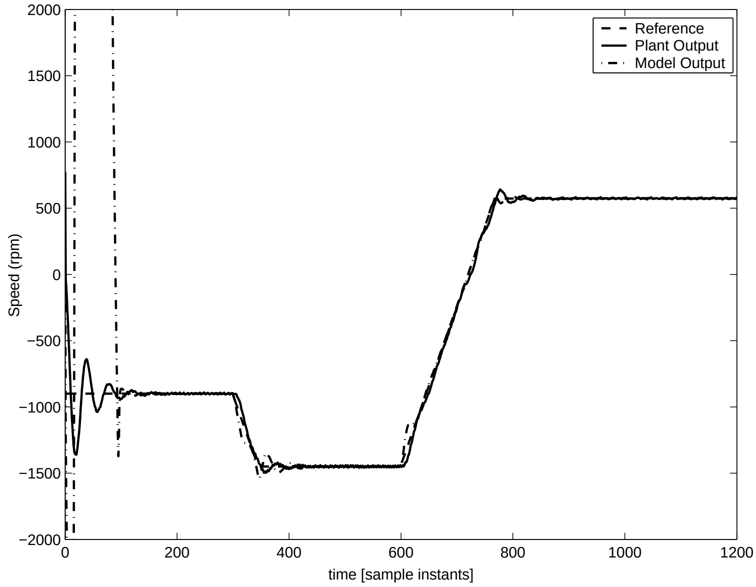

Figure 4.15: Fixed Load: Magnified View of Tracking

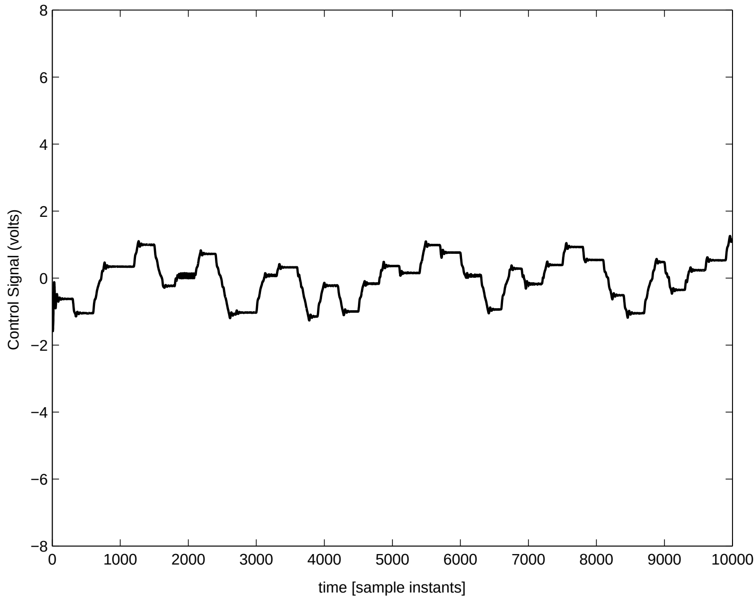

Figure 4.16: Fixed Load: Control Signal

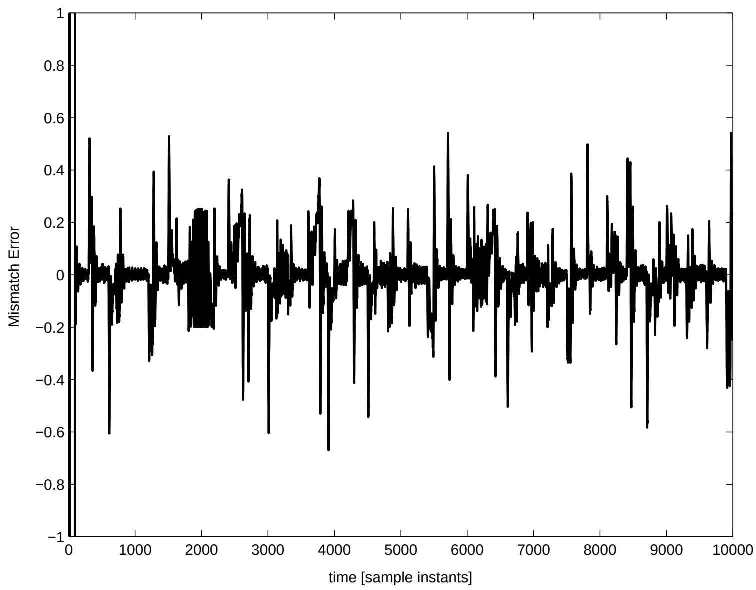

Figure 4.17: Fixed Load: Plant-Model Output Mismatch

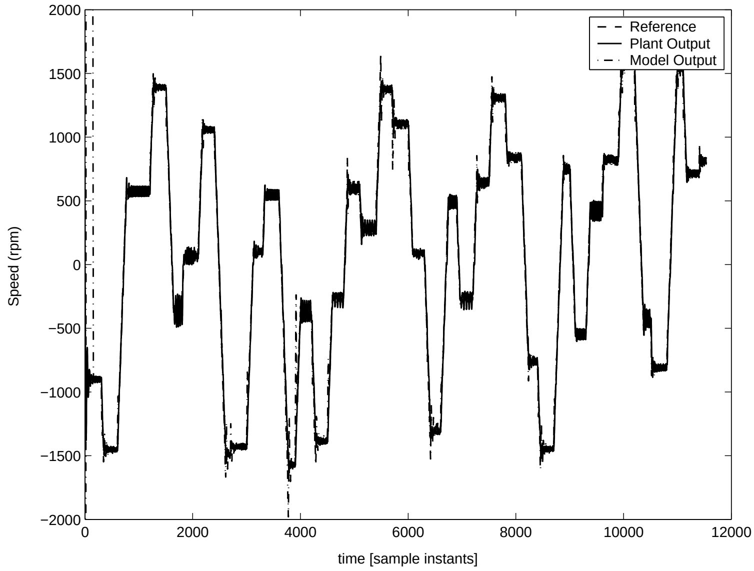

Figure 4.18: Variable Load: Tracking

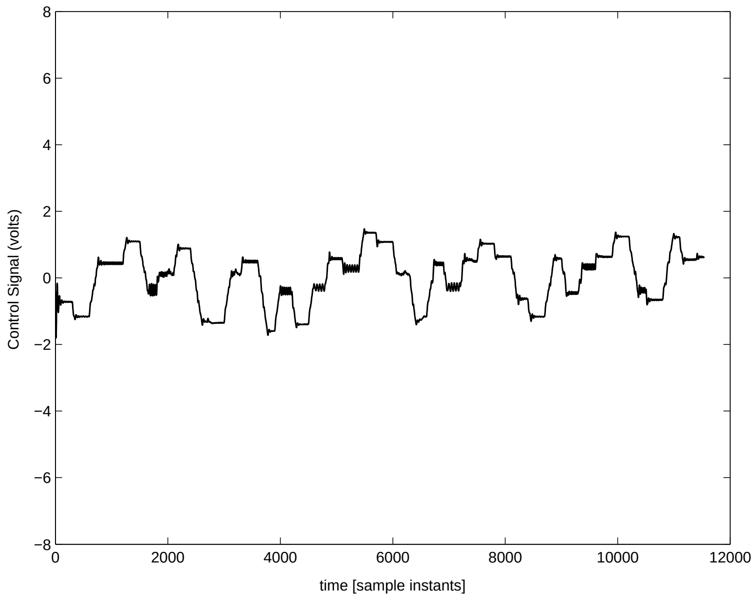

Figure 4.19: Variable Load: Control Signal

CHAPTER 5

A NEW NONLINEAR ADAPTIVE FILTER FOR U-MODEL BASED IMC

In the previous chapter, a simple RBF based nonlinear model was suggested for use in the identification scheme. It is, however, desirable to extend the work to include more general nonlinear adaptive models. These models should provide flexibility in the selection of nonlinear functions and the number and type of inputs to be used. Nonlinear adaptive filtering is a well established theory. By developing a U-Model equivalence of a generalized nonlinear adaptive filter, we can benefit from the rich literature in this area.

5.1 Nonlinear Adaptive Filters

Accurate estimation of the plant is an important part of the IMC scheme. Adaptive filters with nonlinear pre-processing of the inputs provide a versatile structure for the estimation task. It has been shown that nonlinear classifiers have greater capacities than linear classifiers [5]. A typical nonlinear adaptive filter is shown in figure 5.1. Here the input variables$(x_1, x_2, \ldots, x_n)$ are pre-processed through nonlinear functions $\phi_i(.)$. These are then input to the adaptive linear combiner that adaptively assigns weights to these inputs. The resulting sum gives the approximated output which can be written as

$$y_M = \sum_{i=1}^{n} w_i \phi_i\left(x_i\right)$$

A proper selection of the nonlinear functions $\phi_i(.)$ is important for the overall performance of the structure. In particular, a polynomial pre-processor offers great simplicity and beauty. Through it one can realize a wide variety of nonlinear functions by adapting only a single set of weights [5]. Several update algorithms have been developed for use with nonlinear adaptive filters. These include iterative error-correction rules such as the Perceptron and α-LMS rules, and iterative gradient-descent procedures such as the μ-LMS and RLS algorithms. Another candidate, the nLLMS algorithm, has been discussed in detail in section 4.2.

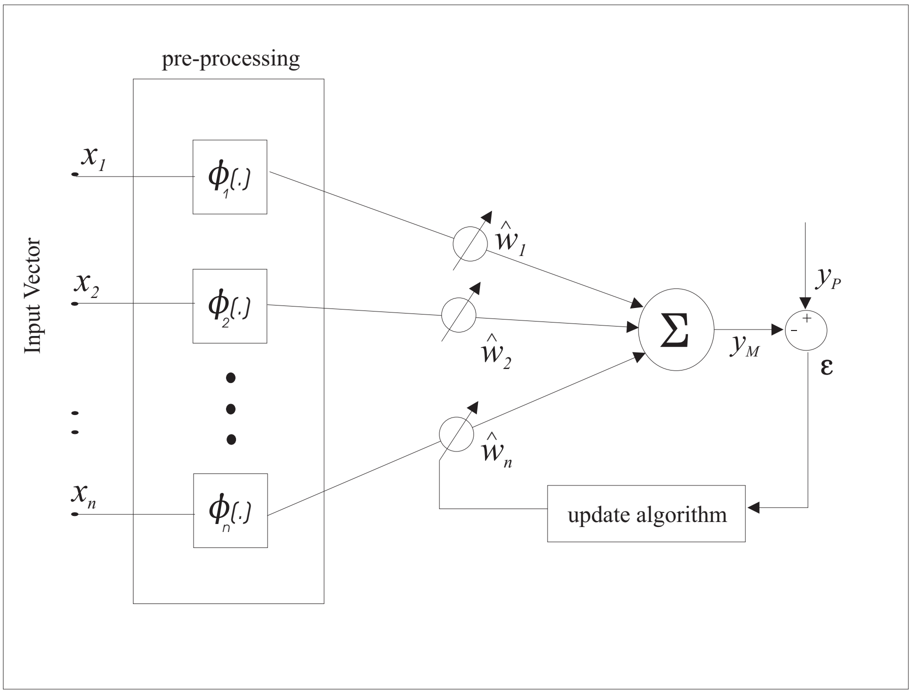

Figure 5.1: Nonlinear Adaptive filter

Our prime objective, in introducing nonlinear adaptive filtering with the U-Model based IMC, is to allow for greater functional approximation capabilities whilst providing design flexibility. This flexibility comes through the choice of nonlinear preprocessors to be used and through the choice of combinations of plant inputs and outputs to be used for training.

5.2 Proposed Nonlinear Adaptive Filter

The proposed nonlinear adaptive filter is depicted in figure 5.2. Here the model output is given by:

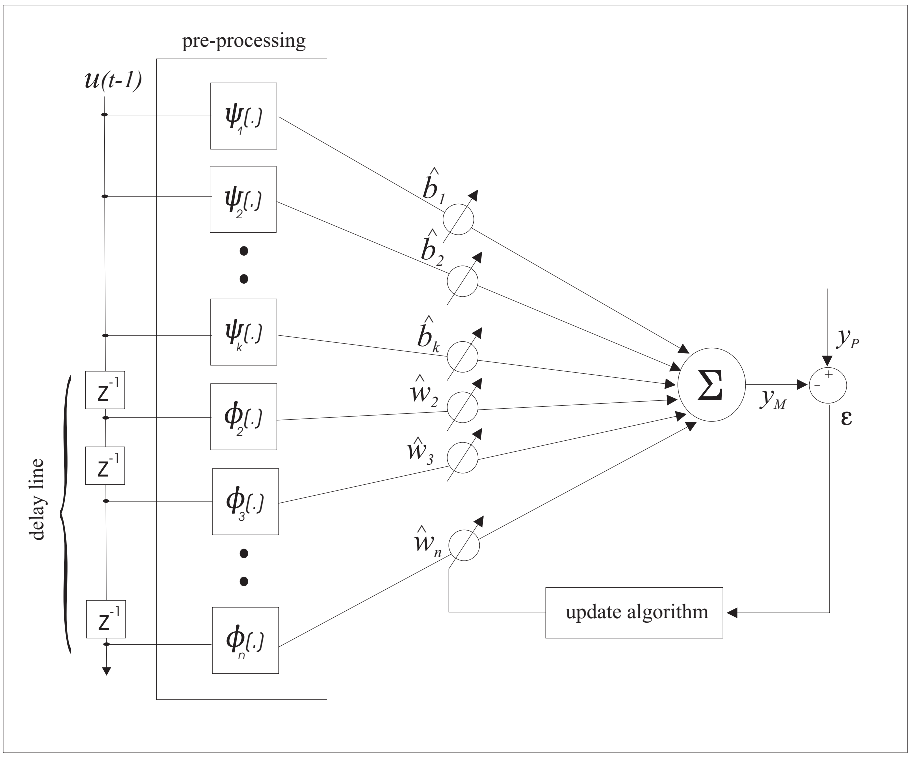

Figure 5.2: Proposed Nonlinear Adaptive filter

$$y_M(t) = \hat{b}_1\psi_1(u(t-1)) + \hat{b}_2\psi_2(u(t-1)) + \ldots + \hat{b}_k\psi_k(u(t-1)) + \hat{w}_2\phi_2(u(t-2)) + \hat{w}_3\phi_3(u(t-3)) + \ldots + \hat{w}_n\phi_n(u(t-n)) \quad (5.1)$$

In order to make this structure suitable for use with the U-Model the functions $\psi_i(.)$ are defined to represent the powers of the current input signal $u(t-1)$. i.e.,

$$\psi_i(u(t-1)) = u^i(t-1) \qquad \text{for} \quad i = 1, 2, \ldots, k \tag{5.2}$$

Additionally, it is proposed to use Gaussian radial basis function (RBF) to represent the remaining nonlinear functions $\phi_j(.)$ in equation (5.1).

$$\phi_j(u(t-j)) = e^{-\left(\frac{u(t-j)-m_j}{c_j}\right)^2} \qquad \text{for} \quad j = 2, 3, \ldots, n \tag{5.3}$$

where m_i and c_i represent the center and spread parameters of the Gaussian RBF, respectively. Radial basis functions are among the most commonly used functions in neural networks and have good approximation capabilities [63]. With these selections the model (5.1) can now be written as:

$$y_M(t) = \hat{b}_1 u(t-1) + \ldots + \hat{b}_k u^k(t-1) + \hat{w}_2 \phi_2(u(t-2)) + \ldots + \hat{w}_n \phi_n(u(t-n)) \tag{5.4}$$

where the functions $\phi_i(.)$ are given by equation (5.3).

This structure has been specially tailored for use with the U-Model, where the first k terms represent powers of the current input and the last $n-1$ terms represent the contributions due to past data. The number of terms required in the model (equation (5.4)) can be adjusted to best suit the plant at hand. Additionally, other nonlinear functions may be used in place of the proposed ones. It is proposed to use the nLLMS algorithm for updating of the weights.

5.3 Using the Proposed Model in U-Model Based IMC

The proposed system block diagram is given in figure 5.3. The main idea is the same as the one detailed in chapter 4. However, we need to present a different U-Model equivalence and control law for this case.

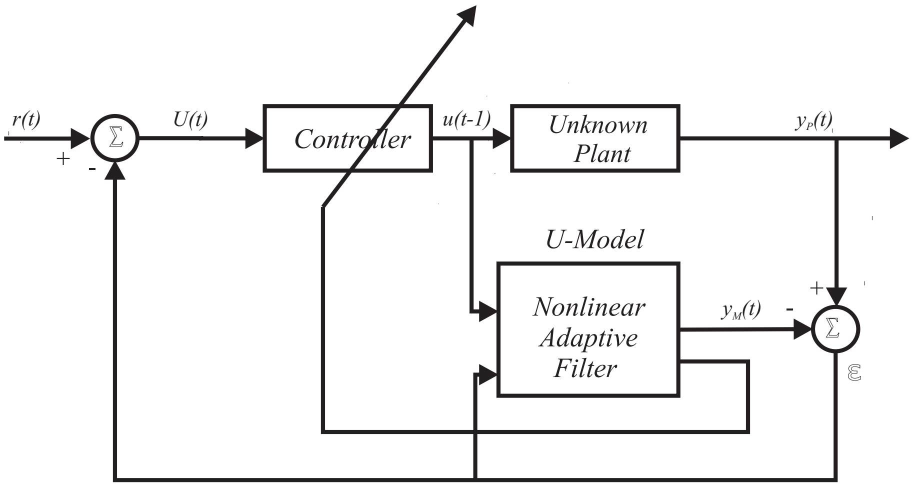

Figure 5.3: Proposed Structure Block Diagram

5.3.1 U-Model Equivalence of the Proposed Model

To allow for the synthesis of a simplistic control law, equation (5.4) is converted into its equivalent U-Model

$$y_M(t) = \sum_{j=0}^{N} \hat{\alpha}_j u^j(t-1) \tag{5.5}$$

where

$$\hat{\alpha}_0 = \hat{w}_2 \phi_2(u(t-2)) + \ldots + \hat{w}_n \phi_n(u(t-n))$$

and

$$\hat{\alpha}_1 = \hat{b}_1, \qquad \hat{\alpha}_2 = \hat{b}_2, \ldots$$

It must be noted the through equation (5.5) we have converted the model into a polynomial in the current input $u(t-1)$. This indicates that the problem of finding the inverse of the plant (to be used as the controller) can be solved using any standard polynomial root-solving algorithm. A large number of root-solving algorithms are available in literature related to numerical methods. Next we discuss one of the most commonly used root-solving methods: the Newton-Raphson algorithm.

5.3.2 Newton-Raphson Algorithm Based Controller for the Proposed Scheme

With reference to the proposed structure block diagram (figure 5.3), we first look at the situation at hand. The input $U(t-1)$ enters the controller and is converted into an output $u(t-1)$, which in turns enters the nonlinear model $f_M(.)$ and is converted into y_M. Now, for controller to be an inverse of the model, we must have $y_M = U$. This leads to the equation:

$$f_M(u(t-1)) = U(t) \tag{5.6}$$

We need to solve this equation for $u(t-1)$. We now develop a simple iterative procedure based on the Newton-Raphson algorithm for solving this nonlinear equation.

Assuming that we begin with an initial root estimate $u_k(t-1)$, we are looking for a new value

$$u_{k+1}(t-1) = u_k(t-1) + h \tag{5.7}$$

such that,

$$f_M(u_{k+1}(t-1)) = U(t) \tag{5.8}$$

is satisfied. To obtain h, we use the first-order Taylor series expansion of the nonlinear function $f_M(.)$ as (assuming that the function is differentiable):

$$f_M(u_{k+1}(t-1)) = f_M(u_k(t-1)) + h\frac{d[f_M(u_k(t-1))]}{d[(u_k(t-1))]} \tag{5.9}$$

Equating (5.8) and (5.9) leads to:

$$h = \frac{f_M(u_k(t-1)) - U(t)}{\frac{d[f_M(u_k(t-1))]}{d[(u_k(t-1))]}} \tag{5.10}$$

Substituting (5.10) in (5.7), gives us the desired update equation as:

$$u_{k+1}(t-1) = u_k(t-1) + \frac{f_M(u_k(t-1)) - U(t)}{\frac{d[f_M(u_k(t-1))]}{d[(u_k(t-1))]}} \tag{5.11}$$

Note that this equation requires the derivative of the function $f_M(.)$ to be known. This might not always be easy. Therefore, an additional advantage of using the U-Model becomes clear here. This added advantage is that since the model itself is a polynomial in the current control term $u(t-1)$, therefore, its derivative with respect to $u(t-1)$ will also be a polynomial in $u(t-1)$. This adds to computational ease of the nonlinear controller. Using the U-Model of equation (5.5) and its derivative:

$$y'_M(t) = \sum_{j=1}^{N} j\hat{\alpha}_j u^{j-1}(t-1) \tag{5.12}$$

leads to the controller:

$$u_{k+1}(t-1) = u_k(t-1) + \frac{\sum_{j=0}^{N} \hat{\alpha}_j {u_k}^j(t-1) - U(t)}{\sum_{j=1}^{N} j\hat{\alpha}_j {u_k}^{j-1}(t-1)} \tag{5.13}$$

Before closing this section, we make two additional comments.

1. In many cases, where it is desired to increase the rate of convergence of the Newton-Raphson algorithm, an additional acceleration factor ω may be employed as:

$$u_{k+1}(t-1) = u_k(t-1) + \omega\frac{\sum_{j=0}^{N} \hat{\alpha}_j {u_k}^j(t-1) - U(t)}{\sum_{j=1}^{N} j\hat{\alpha}_j {u_k}^{j-1}(t-1)} \tag{5.14}$$

2. Since the procedure in equation (5.13) is iterative, we need to employ an initial guess to begin the iterations. A good solution is to use the previously applied control input as the initial value. This is a particulary suitable choice when the signal-to-noise ratio of the plant is high and when the reference input changes are slow [74].

5.3.3 Anticipated Problems

Although the controller structure of equation (5.13) has quite a general appeal, there are two main problems that may occur. First, the denominator in (5.13) may be zero or nearly zero. This can cause problems because, in practice, it cannot be guaranteed that the derivative of the function will not be equal to zero after any particular iteration due to model variation, estimation error and even an unsuitable initial value. Second, it may be that the polynomial has no real roots. To overcome these potential problems while maintaining simplicity, [74]proposed an improved version of the Newton-Raphson algorithm. The detailed procedure can be found in [74]. However, a brief point-wise description is presented here for the sake of completeness. In the following, γ_1 represents the preset tolerance for accepting root solution and γ_2 is a small positive constant.

1. Using the value of $U(t)$ and an initial guess $u_k(t-1)$, evaluate $f_M(u_k(t-1))$ and $f'_M(u_k(t-1))$ using equations (5.5) and (5.12).

2. If $|f_M(u_k(t-1)) - U(t)| \leq \gamma_1$, then $u_k(t-1)$ is the acceptable root. Otherwise, move to the next step.

3. If $f'_M(u_k(t-1))$ is not nearly equal to zero, then all is OK and we can move to the next solution $u_{k+1}(t-1)$ using the update equation (5.13). On the other hand, if $f'_M(u_k(t-1)) \approx 0$ then we need to change the initial guess and restart the procedure. If a fixed number of initial guesses do not yield a solvable problem, then a default value $U(t)/\gamma_2$ can be used to stop the algorithm (in practice, this arrangement will be rarely needed).

With these adjustments, the root-solving controller provides quite a robust performance.

5.4 Application of the Proposed Scheme

In order to demonstrate the effectiveness of the proposed scheme, we carried out simulations as well as real-time experiments. The nonlinear plants used and the results in each case are now discussed in detail.

5.4.1 Adaptive tracking of Hammerstein model

The nonlinear Hammerstein model was introduced in section 4.4.1, here we repeat the model equation for convenience.

$$\begin{aligned} y(t) &= 0.5y(t-1) + x(t-1) + 0.1x(t-2) \\ x(t) &= 1 + u(t) - u^2(t) + 0.2u^3(t) \end{aligned}$$

A model based on the proposed nonlinear filter (5.4) is selected to represent the plant.

$$\begin{aligned} y_M(t) = \hat{b}_1 u(t-1) + \hat{b}_2 u^2(t-1) + \hat{b}_3 u^3(t-1) \\ + \hat{w}_2 \phi_2(u(t-2)) + \hat{w}_3 \phi_3(u(t-3)) \end{aligned} \tag{5.15}$$

The equivalent U-Model (5.5) for this plant is used to synthesize the control law (5.13). An arbitrary reference input is selected and a sampling period of 10ms is used. For the nLLMS, a step size of 0.8 is used and all weights are initialized randomly between 0 and 1. The leakage factor is set to 0.9. The simulation results are depicted in figures 5.4, 5.5 and 5.6. The results suggest that the proposed scheme is able to identify the nonlinear plant online, and the controller provides the appropriate inverse to facilitate tracking of the reference input.

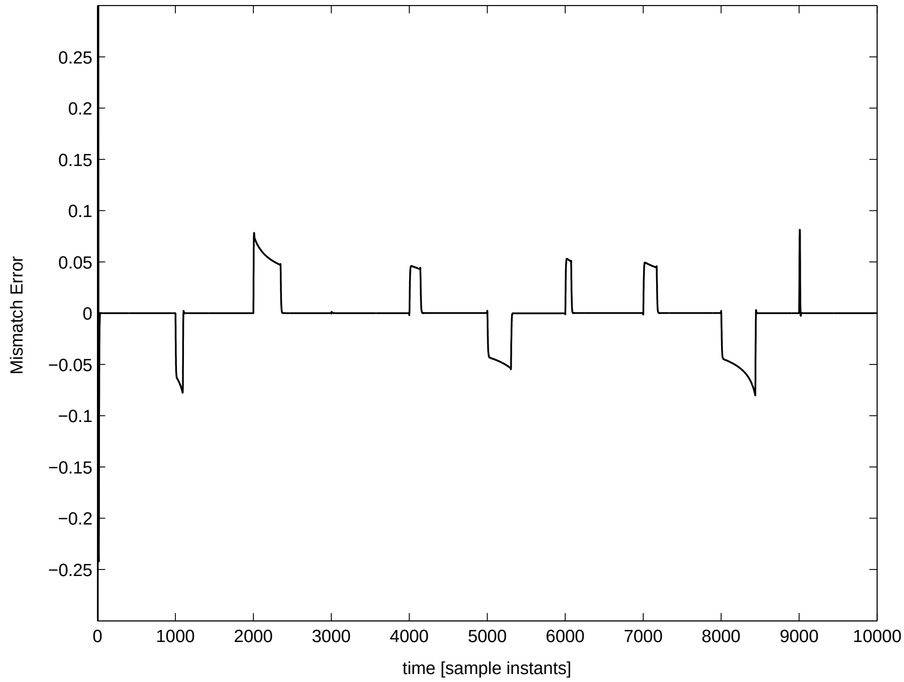

Figure 5.4: Hammerstein Model: Identification error

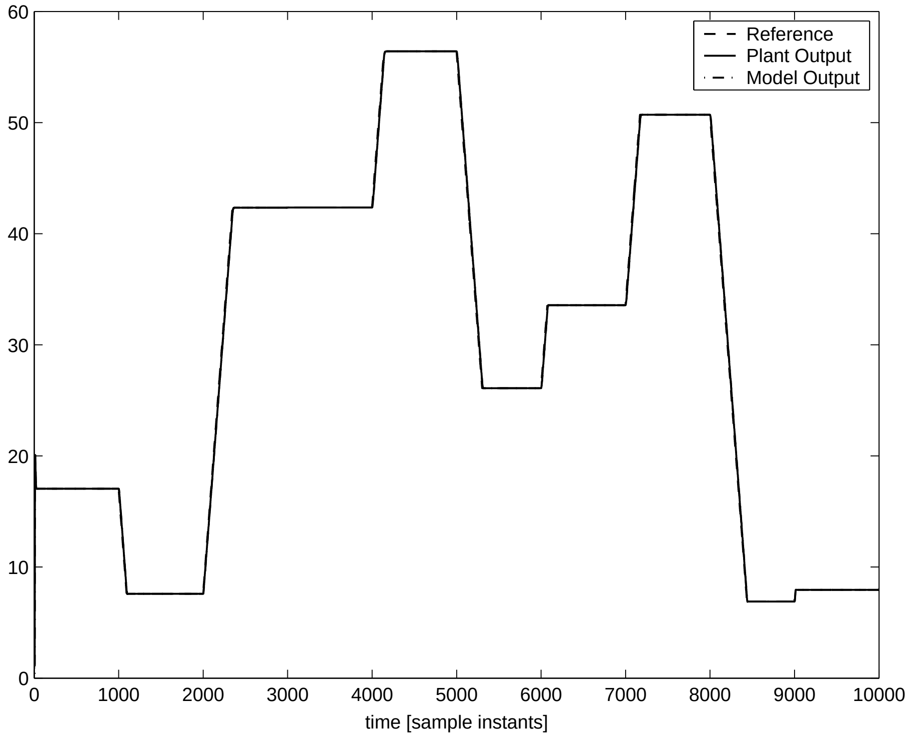

Figure 5.5: Hammerstein Model: Tracking

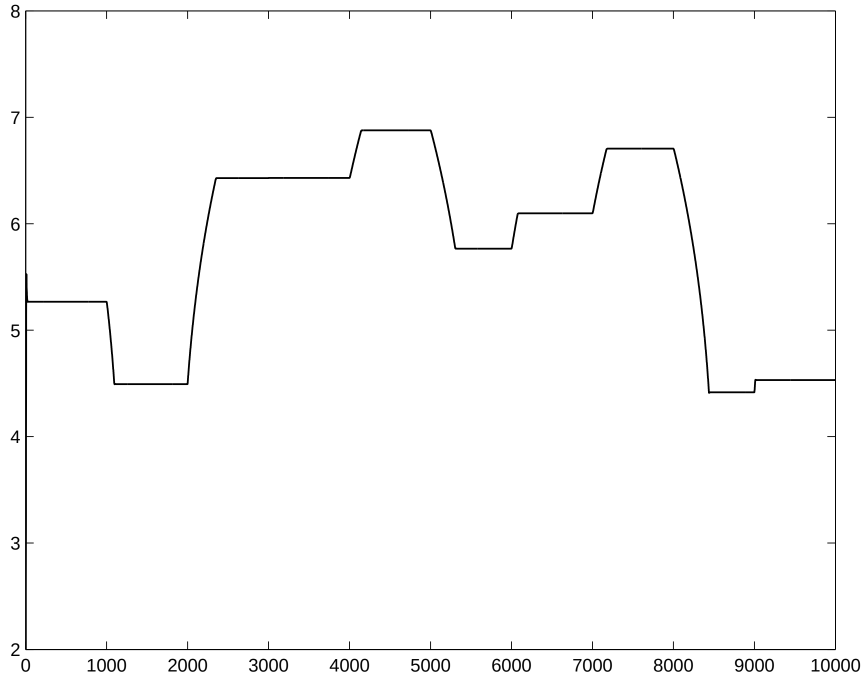

Figure 5.6: Hammerstein Model: Control signal

5.4.2 Adaptive tracking of DC motor speed

The discrete time model used for the simulations was discussed in section 4.4.2.1. The model used to represent the plant is given as:

$$y_M(t) = \hat{b}_1 u(t-1) + \hat{b}_2 u^2(t-1) + \hat{b}_3 u^3(t-1) + \hat{w}_2 \phi_2(u(t-2)) + \hat{w}_3 \phi_3(u(t-3)) \quad (5.16)$$

All other simulation parameters were same as the ones in the previous section. This time again fine tracking of arbitrary reference input is achieved as shown in figure 5.7.

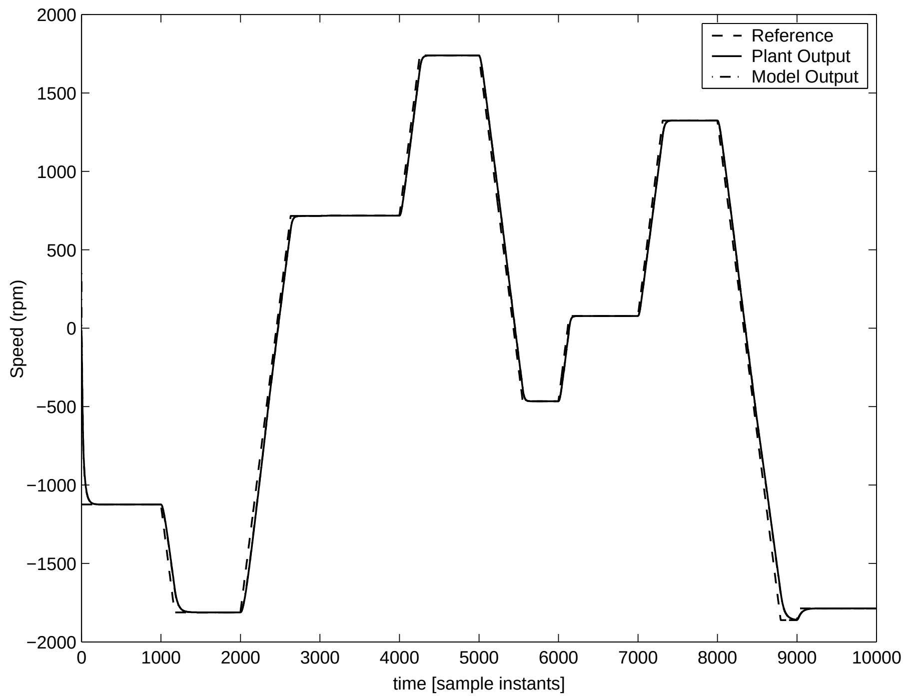

Figure 5.7: DC Motor Speed Tracking

5.4.3 Adaptive tracking of CSTR

A discrete time model for a perfectly mixed, continuously stirred tank reactor (CSTR) is given as [17]:

$$\begin{aligned} y(t+1) = {} & 0.8606y(t) - 0.0401y^2(t) + 0.0017y^3(t) - 0.000125y^4(t) + 0.0464u(t) \\ & - 0.045y(t)u(t) + 0.0034y^2(t)u(t) - 0.00025y^3(t)u(t) - 0.0012u^2(t) \\ & + 0.0013y(t)u^2(t) - 0.0001458y^2(t)u^2(t) + 0.00002083u^3(t) \\ & - 0.00002083y(t)u^3(t) \end{aligned}$$

We now apply the new scheme to this nonlinear plant. The proposed nonlinear adaptive model (5.15) is used to identify the plant and the controller output generated using the Newton-Raphson algorithm (5.13). This time the reference input is chosen to be a periodic triangular wave. A step size of 0.5 is found suitable for this case. Leakage factor is set to 0.9. figures 5.9 and 5.8 show the results.

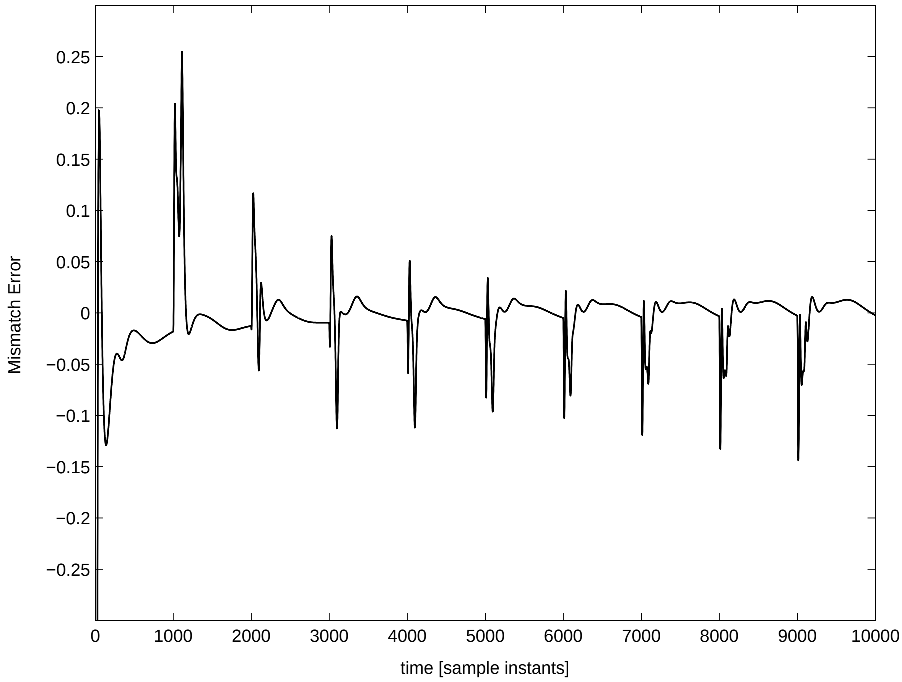

Figure 5.8: CSTR: Identification error

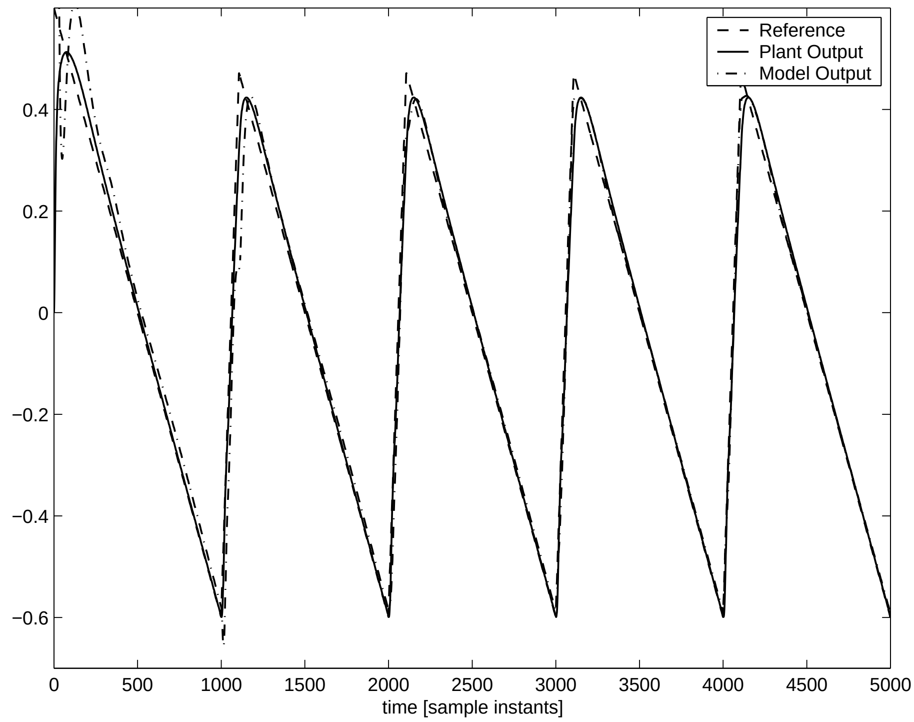

Figure 5.9: CSTR: Tracking

5.4.4 Real-time adaptive tracking of single-link robotic arm manipulator

To further verify the applicability of the proposed scheme, experiments are carried out for the real-time adaptive tracking of a single-link robotic arm position. A block diagram of the experimental setup is shown in figure 5.10, where a DC-motor is used to control the position of a single-link rigid robotic arm. The brush DC-motor (Crouzet 8285002) used has a maximum speed of 3200 revolution per minute, which can be achieved by exciting the motor by 24 volts DC. Standard IBM PC-type Pentium III is used for the computation in real time. Data acquisition is accomplished by Advantech card PCI-1711 and the controller was implemented in SIMULINK real-time windows target environment. The input to the DC-motor is the voltage signal generated by the controller, while the output is the angular position of the robotic arm. This is measured using a potentiometer that gives a voltage signal proportional to the angular position of the robotic arm.

The system is first stabilized using a simple feedback controller. The complete system along with the stabilizing controller is then treated as the plant. An arbitrary reference input is selected and a sampling period of 1ms is used. For the nLLMS, a step size of 0.5 is used and all weights are initialized randomly between 0 and 1. Leakage factor is set t0 0.9. The simulation results are depicted in figures 5.11, 5.12, and 5.13. Figure 5.12 shows the identification error, while figures 5.11 and 5.13 show tracking and the control input respectively. It is evident from this implementation that the proposed scheme is able to properly model the plant and the U-Model based root-solving controller is able to generate appropriate control signals.

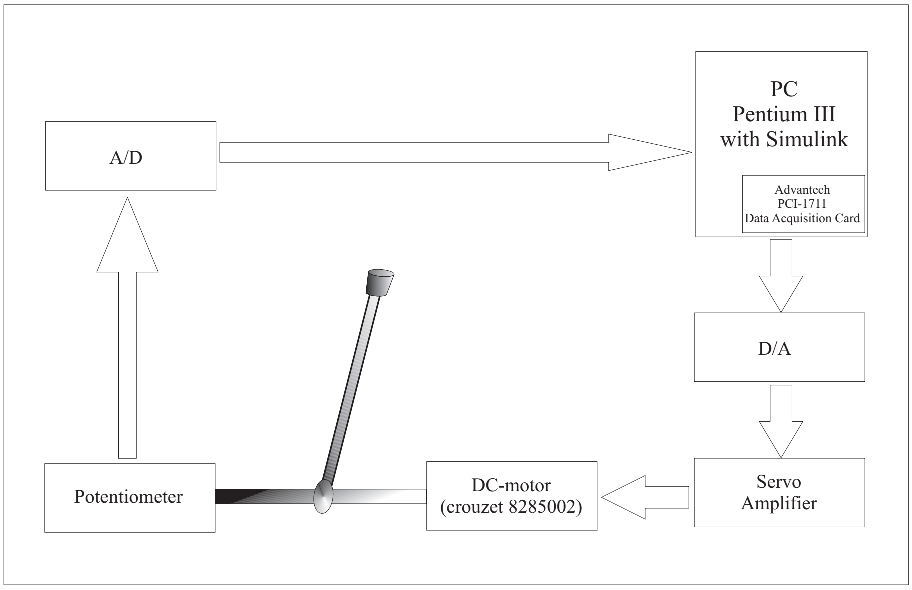

Figure 5.10: Experimental Setup: Block description

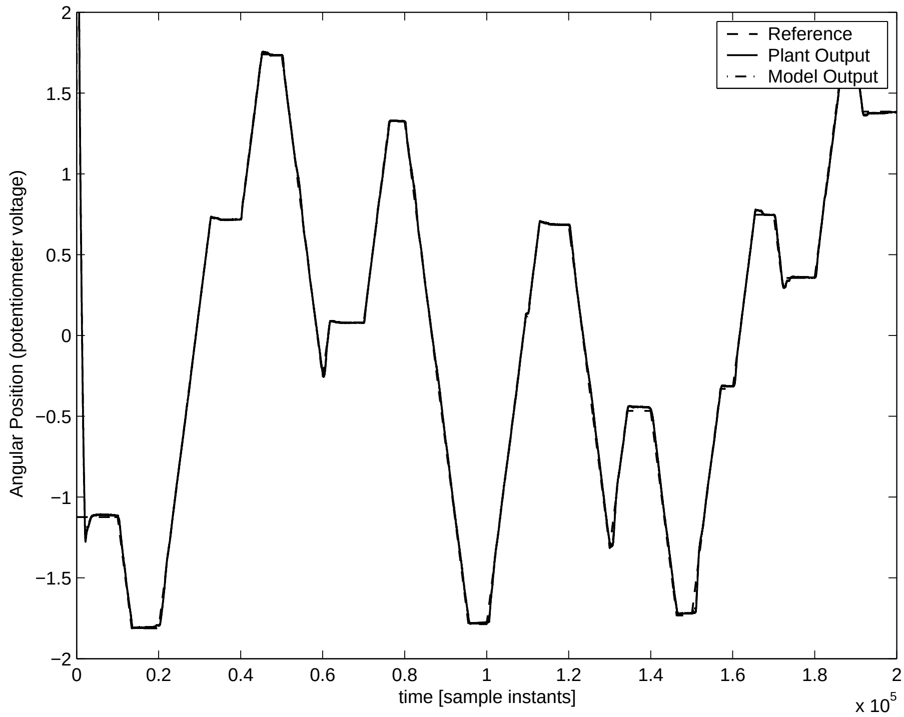

Figure 5.11: Single Link Manipulator: Tracking

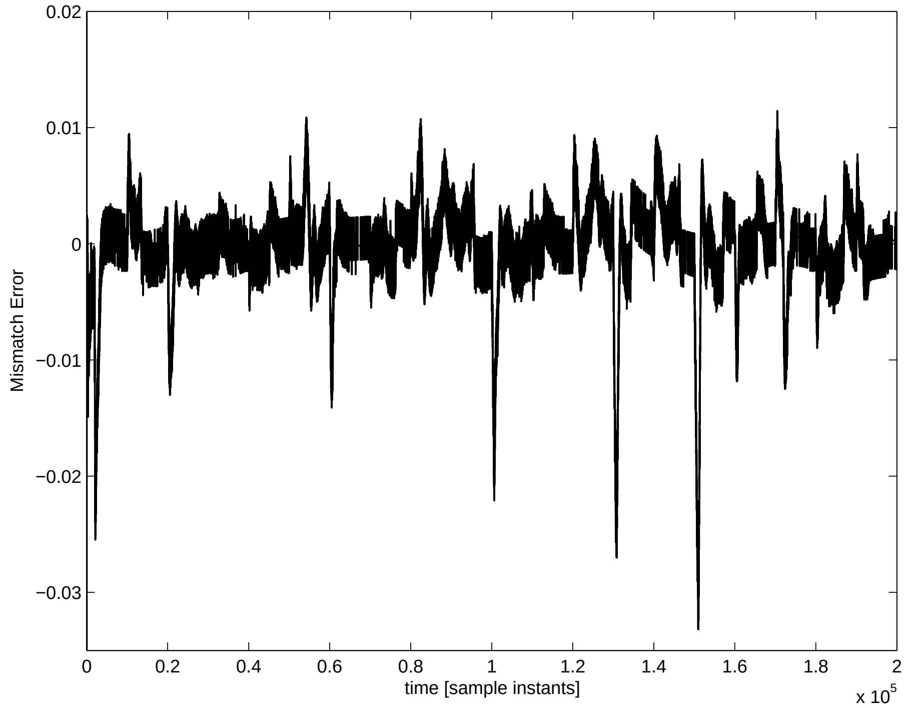

Figure 5.12: Single Link Manipulator: Identification error

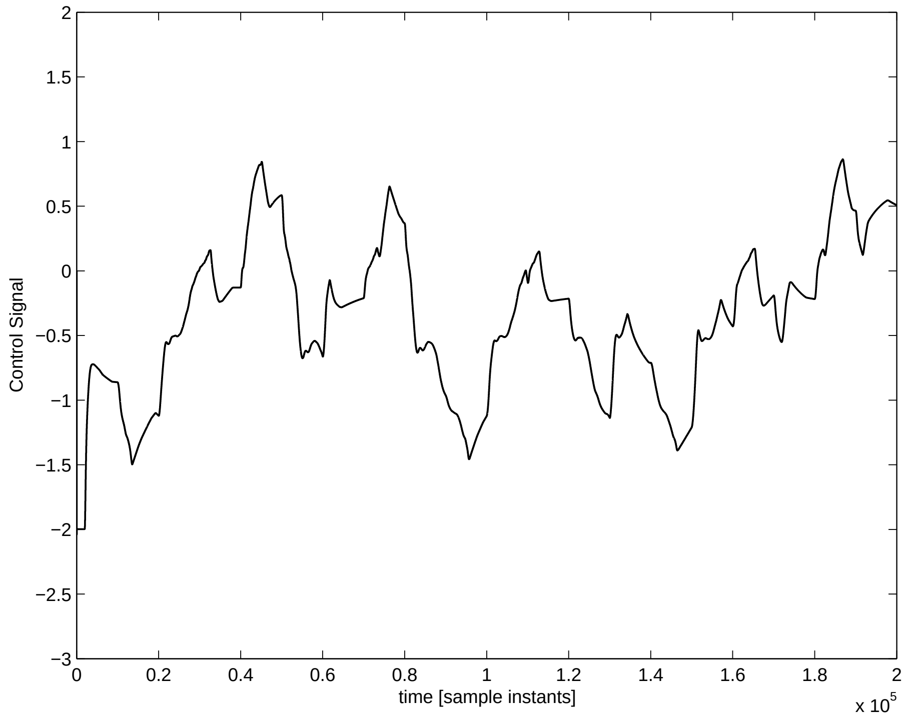

Figure 5.13: Single Link Manipulator: Control signal

CHAPTER 6

HIGHER ORDER NEURAL NETWORKS IN U-MODEL BASED IMC

Neural Networks (NNs) form an important part of Intelligent Control design and have the ability to model a large variety of nonlinear dynamic plants. It is therefore, of interest to utilize NNs in our proposed adaptive scheme. In this chapter we introduce the Higher-Order Neural Networks and propose a U-Model based IMC scheme utilizing these networks. Simulations are presented to demonstrate the use of the proposed method.

6.1 Neural Networks

Neural Networks are parameterized nonlinear functions. Their parameters are, for instance, the weights and biases of the network. Adjustment of these parameters results in different shaped nonlinearities. Typically these adjustments are achieved by a gradient descent approach on an error function that measures the difference between the output of the neural network and output of the actual system. In other words, a neural network is adjusted to serve as an approximator for an unknown function that is only known by how it specifies output values for the given input values. Additionally there is no restriction on the unknown function to be linear. In this way, neural networks provide a logical extension to create nonlinear adaptive control schemes. Today, Neural Networks are being widely used in a number of nonlinear control applications including Model Predictive Control (MPC) , Internal Model Control (IMC) and Adaptive Control. Detailed discussions on the variety of available NNs and their application to control problems can be found in [59] and [63].

6.2 Higher-Order Neural Networks

Among a vast variety of available NNs, Higher-order Neural Networks (HONNs) are particulary suited for capturing the higher-order nonlinear properties of the input pattern space [63] . HONNs are the result of nearly two decades of extensive attempts towards developing architectures of neurons that are capable of capturing not only the linear correlation between the components of the input pattern but also the higher-order correlation between the components of the input patterns. These have been proved to have good computational, storage, pattern recognition, and learning properties and are realizable in hardware [75]. Higher-Order Neural Units (HONUs) are the basic building block of the HONNs. For a typical HONU the output is given by (see figure 6.1)

$$y = \phi(z) \tag{6.1}$$

$$\begin{aligned} z =& w_0 + \sum_{i_1}^{n} w_{i_1} x_{i_1} + \sum_{i_1,i_2}^{n} w_{i_1 i_2} x_{i_1} x_{i_2} + \dots \\ &+ \sum_{i_1,\dots,i_N}^{n} w_{i_1 \dots i_N} x_{i_1} \dots x_{i_N} \end{aligned} \tag{6.2}$$

where $\mathbf{x} = [x_1 \quad x_2 \dots x_n]^T$ is the vector of neural inputs, y is an output, and $\phi(.)$ is an activation function. The HONUs may be used in conventional feed-forward Neural Network structures as hidden units to form HONNs. In this case, however, consideration of the higher correlation may improve the capabilities of the approximation

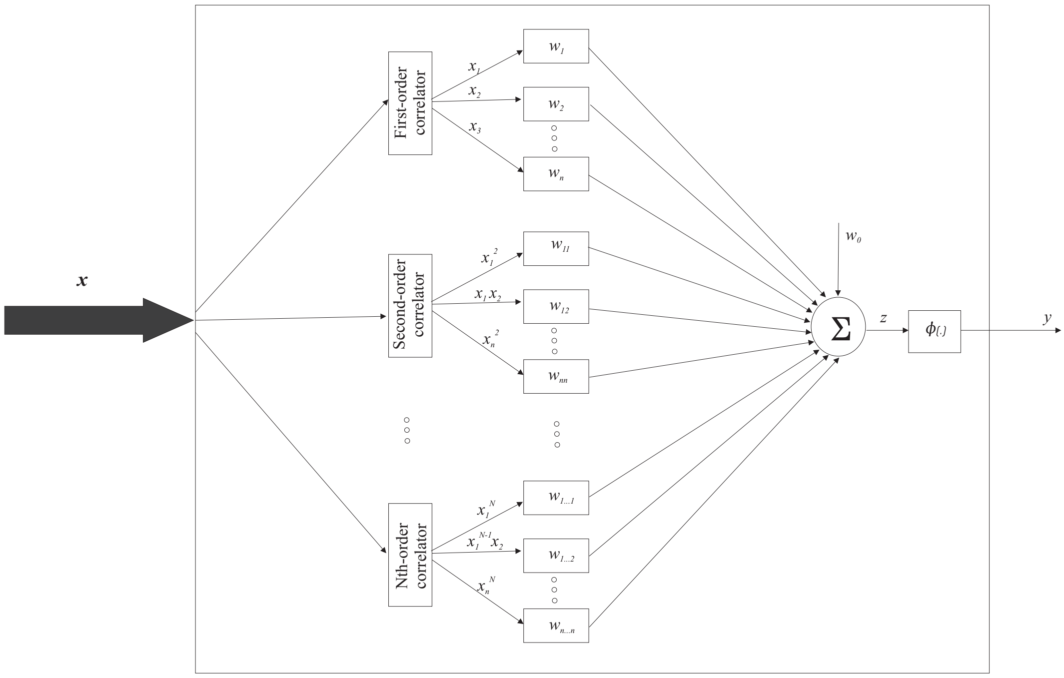

Figure 6.1: A Higher-Order Neural Unit

and generalization of the NNs. Typically only second-order networks are usually employed in practice to give a tolerable number of weights. On the other hand, if the order of HONU is high enough, then, as is known from Stone-Weiestrass theorem, equations (6.1) and (6.2) may be considered as a network with n inputs and a single output. This structure can be treated as a two-layered Neural Network and is capable of dealing with the problems of functional approximation and pattern recognition (see [63] for details).

To accomplish an approximation task for given input-output data $\{\mathbf{x}(k), y(k)\}$, the learning algorithm for the HONN can be easily developed on the basis of the gradient descent method. Let the error function be formulated as

$$E(k) = \frac{1}{2}[d(k) - y(k)]^2$$

where $d(k)$ is the desired output and $y(k)$ is the output of the Neural Network. Minimization of the error function by a standard steepest-descent algorithm yields the following set of learning equations

$$w_0{}^{new} = w_0{}^{old} + \eta(d - y)\phi'(z)$$
$$w_{ij}{}^{new} = w_{ij}{}^{old} + \eta(d - y)\phi'(z)v_{i_1 \ldots i_j} \tag{6.3}$$

where $\phi'(z) = \frac{d\phi}{dz}$ and $v_{i_1 \ldots i_j}$ denote the higher-order correlation terms of the n-dimensional input as:

$$v_{i_1} = x_{i_1} \tag{6.4}$$
$$v_{i_1 i_2} = x_{i_1} x_{i_2}, \qquad 1 \leq i_1, i_2, \ldots, i_N \leq n \tag{6.5}$$
$$v_{i_1 i_2 \ldots i_N} = x_{i_1} x_{i_2} \ldots x_{i_N} \tag{6.6}$$

6.3 U-Model Based Adaptive IMC Using HONNs

This section proposes a new IMC scheme based on HONNs and their equivalent U-Model for tracking of uncertain, stable, nonlinear dynamic plants. As shown in figure 6.2, the output of the controller $u(t)$ is fed to both the unknown plant and the HONN model. The mismatch error ε input to the filter is the difference between the output of the plant $y_P(t)$ and the output of the HONN model $y_M(t)$ .The network parameters are updated using equation (6.3) such that the error ε is minimized. A copy of the HONN parameters (which are the also the parameters of the equivalent U-Model) is fed to the controller online and the controller calculates the inverse of the unknown plant using the Newton-Raphson method based on the U-Model of the plant. If the plant to be controlled is unstable then it is first stabilized using known control strategies.

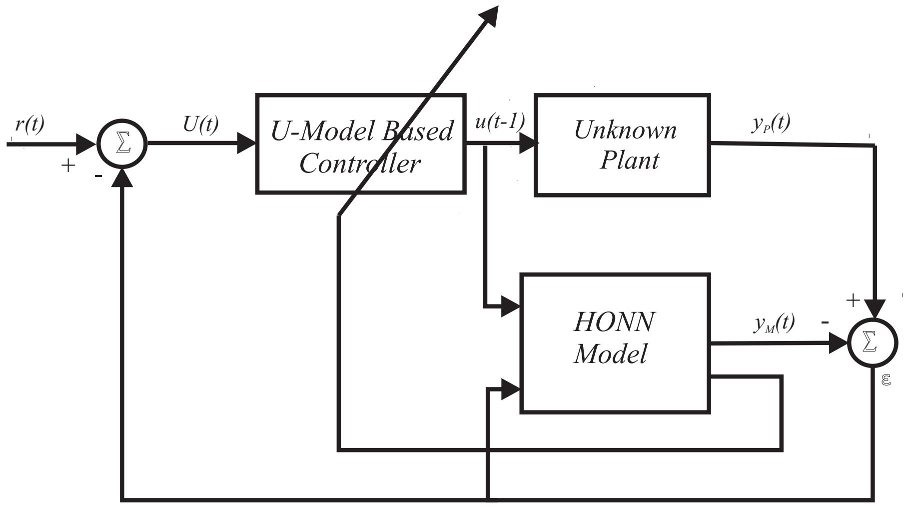

Figure 6.2: Proposed Structure Block Diagram

6.3.1 The Plant

The scheme assumes a stable nonlinear dynamic plant whose functional parameters or the functional structure need not be known.

6.3.2 The Identifying Model

It is proposed to identify the plant online using the HONU depicted in figure 6.1 and given by equations (6.1)and (6.2). As discussed in section 6.2, if the order of HONU is high enough, equations (6.1) and (6.2) may be considered as a network with n inputs and a single output. This structure can be treated as a two-layered Neural Network and is capable of dealing with the problems of functional approximation and pattern recognition [63]. The input vector $\mathbf{x}$ can be composed of combinations of previous inputs and outputs and the exact format can be selected to best suit the plant at hand. For instance, if a third-order HONU is selected ($N = 3$) with three inputs

($\mathbf{x} = [x_1 \quad x_2 \quad x_3]^T$), the NN model can be written as:

$$\begin{aligned} y_M =& \phi \big(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_{11} {x_1}^2 + w_{12} x_1 x_2 \\ & + w_{13} x_1 x_3 + w_{22} {x_2}^2 + w_{23} x_2 x_3 + w_{33} {x_3}^2 + w_{111} {x_1}^3 \\ & + w_{112} {x_1}^2 x_2 + w_{113} {x_1}^2 x_3 + w_{122} x_1 {x_2}^2 + w_{123} x_1 x_2 \\ & x_3 + w_{133} x_1 {x_3}^2 + w_{222} {x_2}^3 + w_{223} {x_2}^2 x_3 + w_{233} x_2 {x_3}^2 \\ & + w_{333} {x_3}^3 \big) \end{aligned} \tag{6.7}$$

Here we can select $\phi(z) = z$ and the weights can be updated online using update equation (6.3). The main idea is to utilize the power of HONNs to capture the higher-order nonlinear properties of the input pattern space (the input to the HONU in this case consists of past inputs and outputs of the nonlinear plant.)

6.3.3 The Control Law

To allow for the synthesis of a simplistic control law, the NN model of equation (6.2) is converted into its equivalent U-Model. This is done by first selecting $\phi(z) = z$ and then expanding the summation terms and regrouping to form a power series in the current control term. The equivalent U-Model will clearly vary according to the selection of the HONU order and according to the definition of the input vector $\mathbf{x}$. For instance, if (6.7) is selected as the model in figure 6.2 and the neural input vector is select as

$$\mathbf{x} = [x_1 \quad x_2 \quad x_3]^T = [u(t-1) \quad u(t-2) \quad u(t-3)]^T$$

(where $u(t)$ represents the input to the plant at discrete time t), then the equivalent U-Model for (6.7) can be written as:

$$y_M(t) = \alpha_0(t) + \alpha_1(t) u(t-1)$$

$$+ \alpha_2(t)u^2(t-1) + \alpha_3(t)u^3(t-1) \tag{6.8}$$

with

$$\alpha_0(t) = w_0 + w_2u(t-2) + w_3u(t-3) + w_{22}u^2(t-2) + w_{23}$$
$$u(t-2)u(t-3) + w_{33}u^2(t-3) + w_{222}u^3(t-2) + w_{223}$$
$$u^2(t-2)u(t-3) + w_{233}u(t-2)u^2(t-3) + w_{333}u^3(t-3)$$

$$\alpha_1(t) = w_1 + w_{12}u(t-2) + w_{13}u(t-3) + w_{122}u^2(t-2)$$
$$+ w_{123}u(t-2)u(t-3) + w_{133}u^2(t-3)$$

$$\alpha_2(t) = w_{11} + w_{112}u(t-2) + w_{113}u(t-3)$$

$$\alpha_3(t) = w_{111}$$

With this structure, the control inputs $u(t-1)$ to the nonlinear plant of figure 6.2 can be easily obtained using the Newton-Raphson polynomial root-solving algorithm discussed in detail in chapter 5. It must be noted that the proposed scheme leads to a very simple and general control law. This approach is therefore expected to prove extremely useful in the area of nonlinear control.

6.4 Application To Nonlinear Plants

To demonstrate the application of the proposed scheme, simulations were carried out for the adaptive tracking of the Hammerstein model and DC motor speed. All programs were run using the SIMULINK platform. This section presents the results of these simulations.

6.4.1 Adaptive tracking of Hammerstein model

The Hammerstein model is used as the unknown plant. It is modelled according to (6.7) and its equivalent U-Model (6.8) is used to convert the control law into a simple root-solving routine (5.13). An arbitrary reference input is selected and a sampling period of 10ms is used. For the update equations, a step size of 0.3 is used and all weights were initialized randomly between 0 and 1. Leakage factor is chosen to be 1. The simulation results are depicted in figures 6.3, 6.4 and 6.5. The results suggest that the proposed scheme was able to identify the nonlinear plant online, and the controller provided the appropriate inverse to facilitate tracking of the reference input.

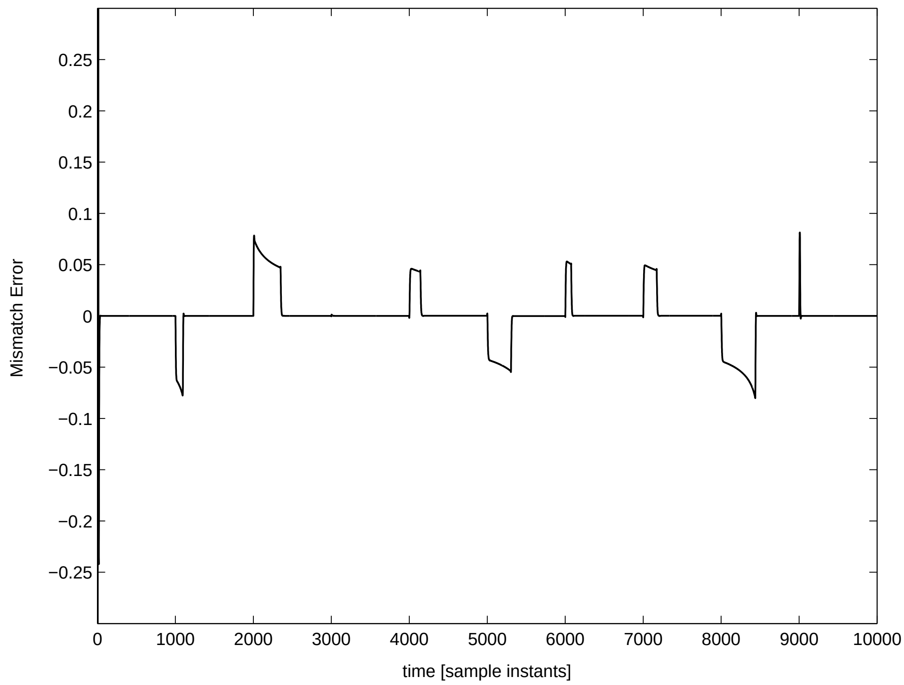

Figure 6.3: Hammerstein Model: Identification error

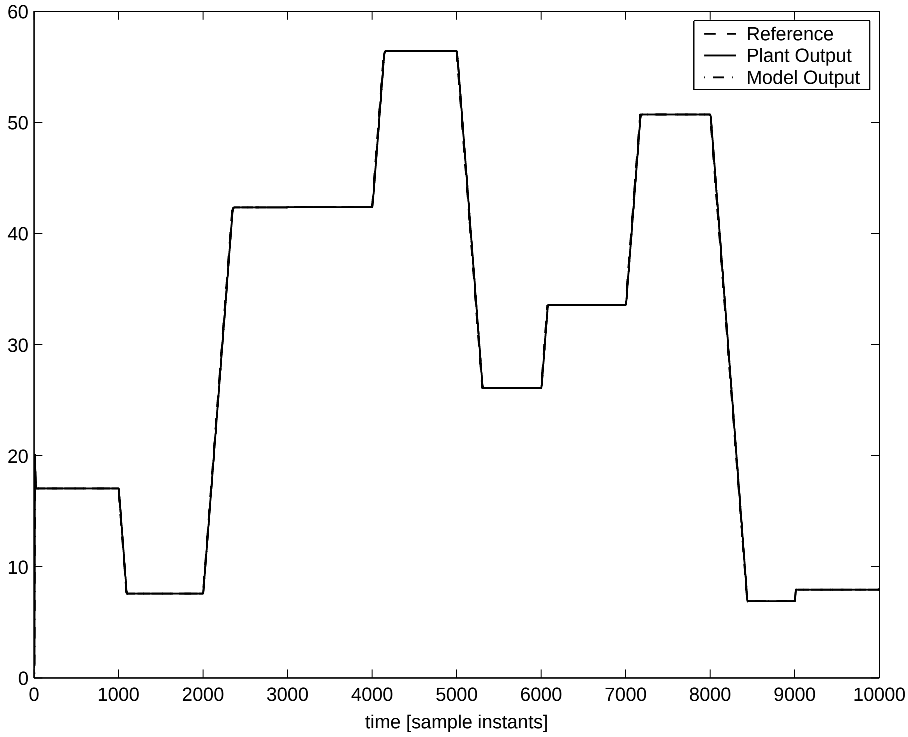

Figure 6.4: Hammerstein Model: Tracking

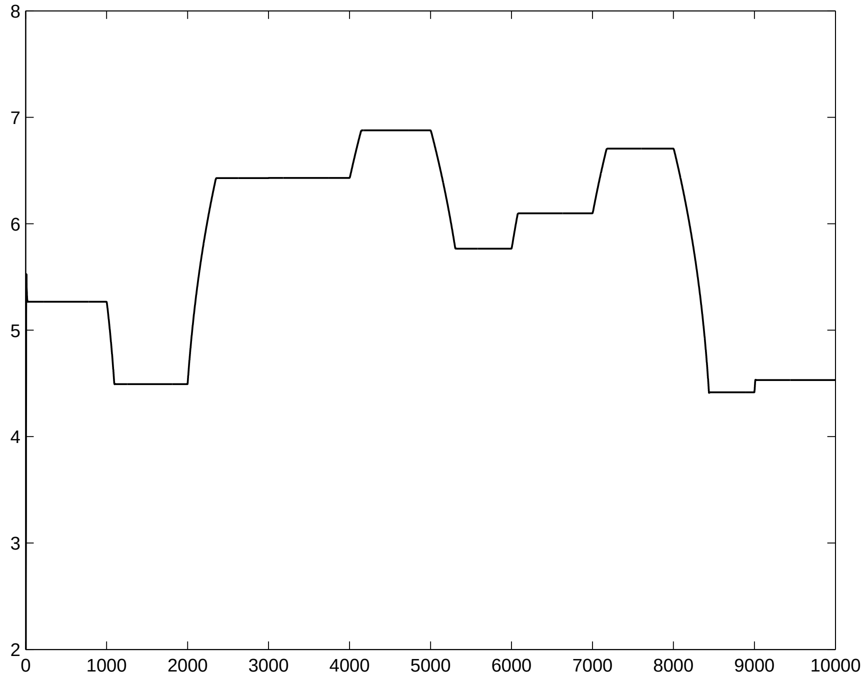

Figure 6.5: Hammerstein Model: Control Signal

6.4.2 Adaptive tracking of DC Motor Speed

For this experiment the DC motor nonlinear model introduced in section 4.4.2.1, is used. The resulting plots are given in figures 6.6, 6.7, and 6.8.

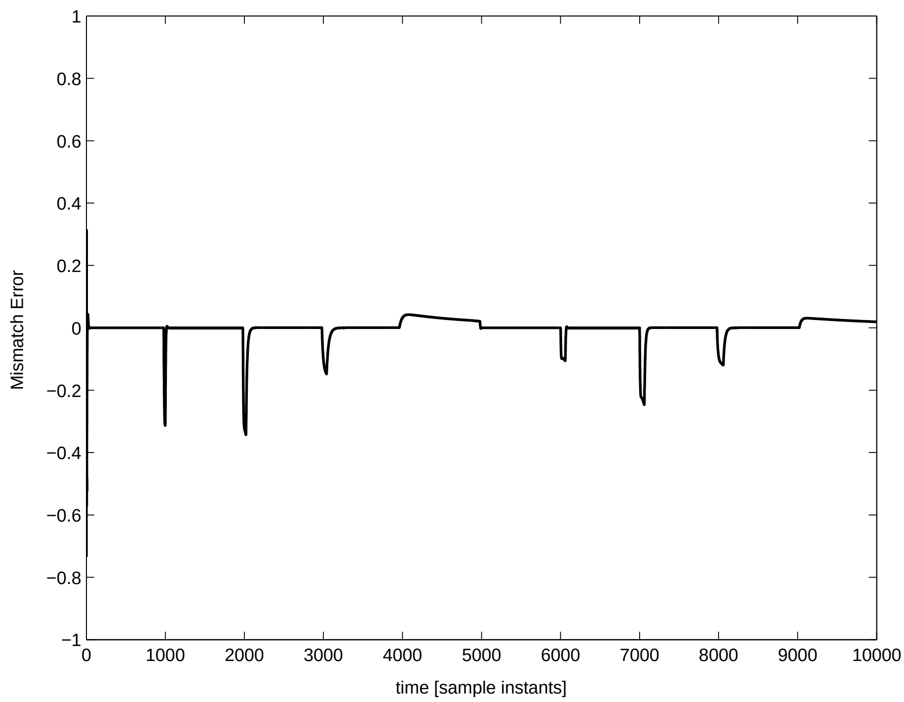

Figure 6.6: Plant-Model Output Mismatch

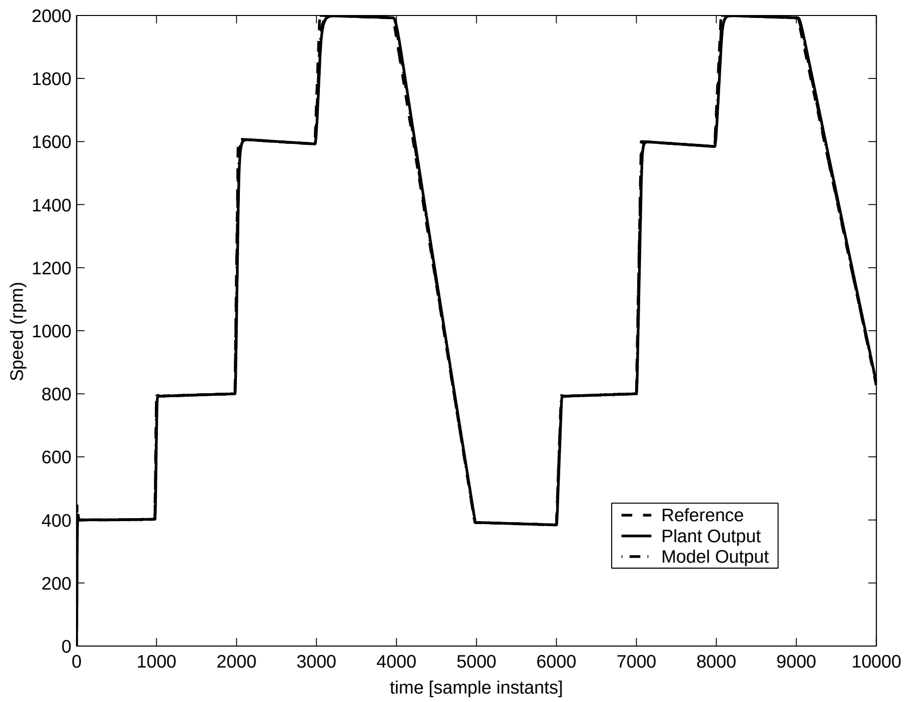

Figure 6.7: DC Motor Speed Tracking

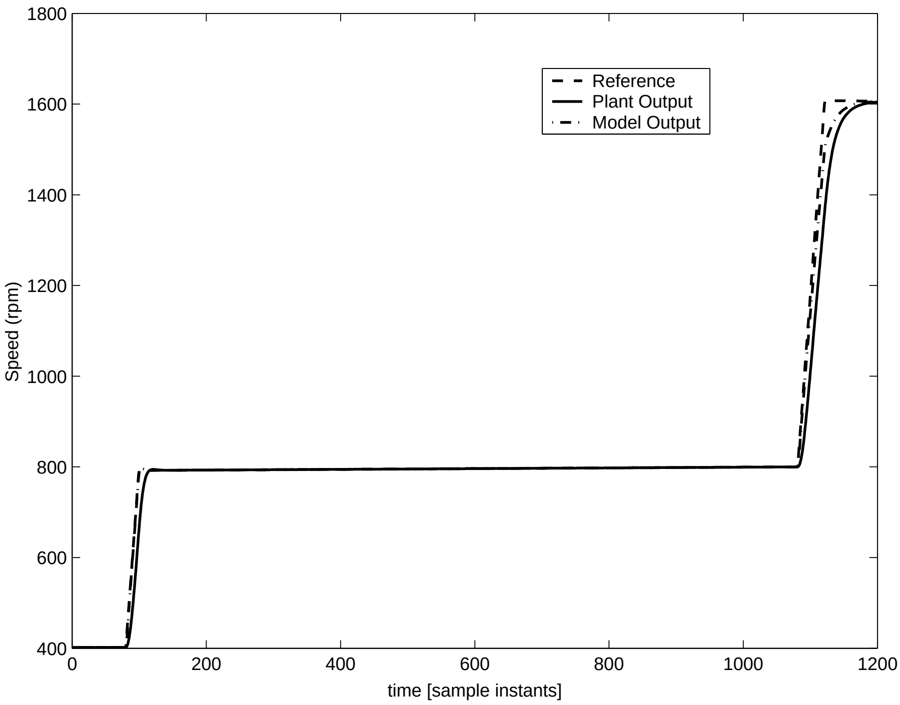

Figure 6.8: Magnified View of Tracking

6.4.3 Comparison with Nonlinear Adaptive Backstepping Controller

The proposed intelligent controller is also compared with the most "systematic" design approach available in nonlinear adaptive control, namely: backstepping [76]. The idea of backstepping is briefly explained with the help of a simple example. Consider the nonlinear plant

$$\dot{y} = u + \varphi(y)^T \theta$$

where $\varphi(.)$ and θ represent the nonlinearity vector and the parameter vector, respectively. The nonlinear adaptive backstepping controller for output tracking can be given as (with y_d as the desired output trajectory)

$$u = c(y_d - y) + \varphi(y_d - y)^T \hat{\theta}$$

with parameter estimator

$$\dot{\hat{\theta}} = \Gamma \varphi(y_d - y)(y_d - y)$$

Here c is a positive constant while Γ represents a positive definite symmetric matrix referred to as the "adaptation gain". Note that although this controller adaptively estimates the parameter vector θ, it does require the exact knowledge of the nonlinearity function $\varphi(.)$. On the other hand, the proposed intelligent controller performs the adaptive tracking of such a nonlinear plant without any knowledge of the nonlinearity function or the parameter vector. The nonlinear plant used for the comparison is

$$\dot{y} = u - 0.1log(|y + 0.1|) - 0.1y^2$$

The system was run in SIMULINK with a sampling time of 1 ms. The desired trajectory was set as a slowly varying arbitrary signal. The results are shown in figures 6.9 and 6.10. The mean square error obtained by the backstepping controller was 0.145 while that achieved by the proposed intelligent controller was 2.13. It is clear that the proposed scheme is capable of tracking the plant without requiring the knowledge of the nonlinearity at the cost of a slightly higher mean square error.

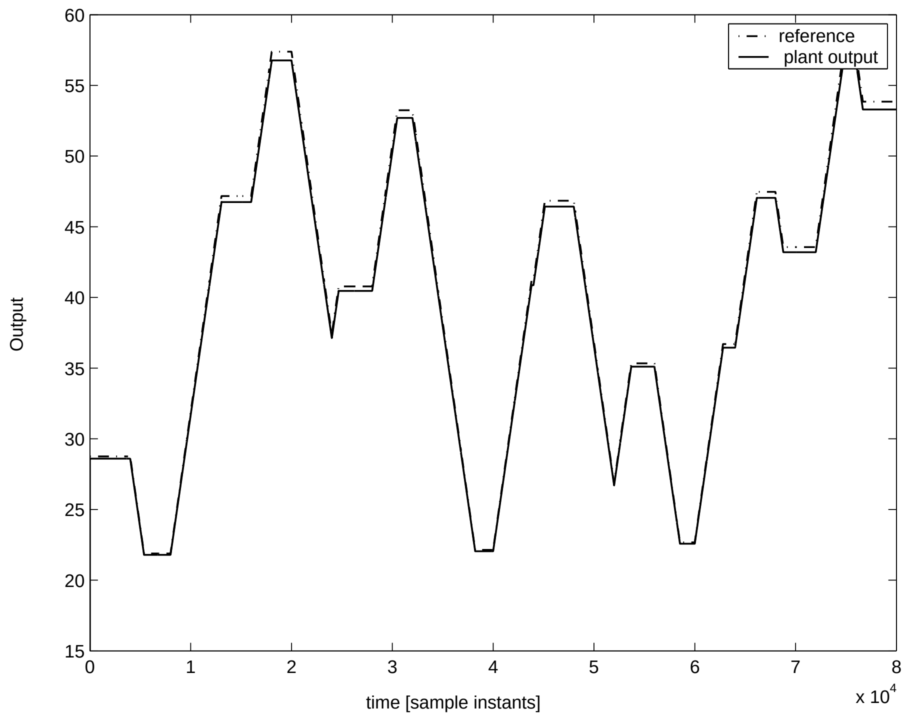

Figure 6.9: Tracking with Backstepping Controller

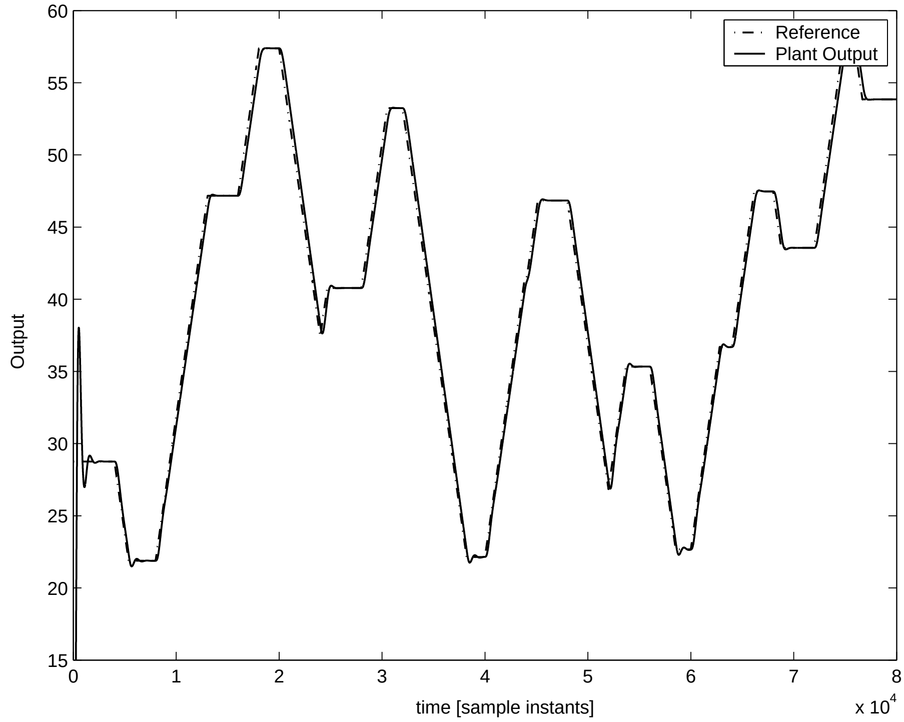

Figure 6.10: Tracking with proposed Intelligent Controller

CHAPTER 7

CONCLUSIONS, WORK SUMMARY & RECOMMENDATIONS

7.1 Conclusions

A novel technique based on the control-oriented U-Model for the adaptive tracking of a wide range of stable nonlinear dynamic plants using only input-output data was introduced. The overall scheme was based on the robust internal model control (IMC) structure wherein different internal models, using nonlinear adaptive filtering and higher-order neural networks, were proposed. In each case, the U-Model equivalence of the internal model was developed and a simplistic control law based on polynomial root-solving was synthesized. Stability of the proposed adaptive scheme was analyzed with the help of various analytical tools. The effectiveness of the proposed adaptive schemes was demonstrated through simulations and real-time applications to a variety of nonlinear plants that include: Hammerstein model, CSTR model, DC motor and single-link robotic arm manipulator.

It is noted that the proposed work was able to combine the robustness of the IMC and the control-oriented nature of the U-Model with the approximation capabilities of nonlinear adaptive filters and neural networks to provide a comprehensive nonlinear adaptive control scheme. The contribution is therefore expected to prove extremely useful for practical applications of nonlinear control.

7.2 Work Summary

- A U-Model based adaptive IMC structure was developed for the adaptive tracking of stable single-input-single-output(SISO) nonlinear dynamic plants.
- Three different nonlinear models were proposed for model identification in the adaptive IMC. These are:
  - A Radial Bases Functions (RBF) based nonlinear model
  - A new general polynomial-kind nonlinear model
  - A Higher-Order Neural Network (HONN) based nonlinear model
- For each of the modelling frameworks above, a U-Model equivalence was developed and a controller was synthesized.
- The use of normalized Leaky Least Mean Square algorithm (nLLMS), for the model identification, was proposed and justified in detail.
- Stability analysis of the proposed adaptive IMC scheme was carried out using the small gain theorem and internal stability.
- The use of Newton-Raphson algorithm for controller synthesis in the proposed adaptive IMC was studied and justifications for its applicability were provided.
- Simulations were carried out for the adaptive tracking of a number of nonlinear dynamic plants using the proposed work. These plants are:
  - Hammerstein Model
  - Continuously Stirred Tank Reactor (CSTR) model

  - Permanent Magnet DC-Motor Model

- Real-time implementation of the proposed adaptive IMC to the adaptive tracking of laboratory-scale DC-Motor speed using SIMULINK platform.

- Real-time implementation of the proposed adaptive IMC to the adaptive tracking of two nonlinear plants that are initially unstable and need to be stabilized first using feedback. These are:

  - Permanent Magnet DC-Motor position tracking

  - Single-link robotic arm manipulator position control

7.3 Recommendations for Further Work

- Further analysis of the proposed schemes should be carried out to bring out their comparative advantages.

- The proposed schemes can be extended to Multi-Input-Multi-Output systems.

- Extensions should me made to include other kinds of neural network structures.

- The present schemes work only for stable plants, a similar adaptive IMC needs to be developed that can perform stabilization as well. Such a contribution would indeed be phenomenal.

- Extensions to two-degree-of-freedom IMC can be investigated.

- Use of root-solving procedures other than the Newton-Raphson algorithm can be explored for possible improvements in the control signals.

REFERENCES

[1] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 8th ed. CA: Addison Wesley Longman, Inc., 1998.

[2] C. Chen, *Linear System Theory and Design*, 3rd ed. NY: Oxford University Press, Inc., 1999.

[3] J. R. Leigh, *Control Theory*, 2nd ed., ser. Control Series 64, D. Atherton, G. W. Irwin, and S. Spurgeon, Eds. London: IEE, 2004.

[4] J. T. Spooner, M. Maggiore, R. Ordonez, and K. M. Passino, *Stable Adaptive Control And Estimation For Nonlinear Systems*. NY: Wiley-Interscience, 2002.

[5] B. Widrow and P. Walach, *Adaptive Inverse Control*. NJ: Prentice-Hall, Inc., 1996.

[6] M. Shafiq and S. H. Riyaz, "Internal model control structure using adaptive inverse control strategy." in *The 4th Int. Conf. on Control and Automation (ICCA)*, 2003, p. 59.

[7] K. J. Hunt and D. Sbarbaro, "Neural networks for nonlinear internal model control," *Proc. Inst. Elect. Eng.*, vol. 138, no. 5, pp. 431–438, 1991.

[8] K. Hunt and D. Sbarbaro, "Adaptive filtering and neural networks for realisation of internal model control," *Intelligent Systems Engineering*, vol. 2, no. 2, pp. 67 – 76, Summer 1993.

[9] J. Sousa, R. Babuska, and H. Verbruggen, "Internal model control with a fuzzy model: application to an air-conditioning system," in *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, vol. 1, July 1997, pp. 207–212.

[10] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*, ser. Information and System Science Series, T. Kailath, Ed. NJ: Prentice Hall, Inc., 1989.

[11] C. E. Garcia and M. Morari, "Internal model control.1.a unifying review and some new results," *Ind. Eng. Chem. Process Des. Dev*, vol. 21, pp. 308–323, 1982.

[12] C. G. Economou, M. Morari, and B. O. Palsson, "Internal model control.5.extension to nonlinear systems." *Ind. Eng. Chem. Process Des. Dev.*, vol. 25, no. I, pp. 403–411, 1986.

[13] A. Datta and J. Ochoa, "Adaptive internal model control: Design and stability analysis," *Automatica*, vol. 32, no. 2, pp. 261–266, 1996.

[14] R. K. Pearson, "Selecting nonlinear model structures for computer control," *Journal of Process Control*, vol. 13, pp. 1–26, 2003.

[15] S. Qin and T. Badgwell, "An overview of nonlinear model predicitive control applications," in *International Symposium on Nonlinear Model Predictive Control: Assessment and Future Directions*, F. Allgower and A. Zheng, Eds., Ascona, Switzerland, 1998, pp. 128–145.

[16] J. Lee, "Modelling and identification for nonlinear model predictive control: Requirements, current status and future research," in *International Symposium on*

Nonlinear Model Predictive Control: Assessment and Future Directions, F. Allgower and A. Zheng, Eds., Ascona, Switzerland, 1998, pp. 91–107.

[17] Q. M. Zhu and L. Z. Guo, "A pole placement controller for nonlinear dynamic plants," *J. Systems and Control Engineering*, vol. 216, no. I, pp. 467–476, 2002.

[18] I. J. Leontarities and S. A. Billings, "Input-output parametric models for nonlinear systems. part i: Deterministic nonlinear systems; part ii: Stochastic nonlinear systems," *Int. J. of Control*, vol. 41, no. 2, pp. 303–344, 1985.

[19] L. Piroddi and W. Spinelli, "An identification algorithm for polynomial NARX models based on simulation error minimization." *Int. J. of Control*, vol. 76, no. 17, pp. 1767–1781, 2003.

[20] K. R. Sales and S. A. Billings, "Self-tuning control of nonlinear ARMAX models." *Int. J. of Control*, vol. 51, no. 4, pp. 753–769, January 1990.

[21] M. Shafiq and M. Haseebuddin, "Internal model control for nonlinear dynamic plants using U-Model," in *12th Mediterranean Conference on Control and Automation*, Turkey, June 2004.

[22] J. E. Slotine and W. Li, *Applied Nonlinear Control*. NJ: Prentics Hall, Inc., 1991.

[23] G. Lightbody and G. Irwin, "Nonlinear control structures based on embedded neural system models," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 553 – 567, May 1997.

[24] C. H. Choi and H. C. Kim, "A robust internal model adaptive controller," in *Proceedings of the 36th Conference on Decision and Control*, San Diego, CA, Dec 1997.

[25] C. Kambhampati, R. Craddock, M. Tham, and K. Warwick, "Inverting recurrent neural networks for internal model control of nonlinear systems," in *Proceedings of the American Control Conference*, vol. 2, June 1998, pp. 975 – 979.

[26] M. Hagan and H. Demuth, "Neural networks for control," in *Proceedings of the American Control Conference*, vol. 3, June 1999, pp. 1642 – 1656.

[27] I. Rivals and L. Personnaz, "Nonlinear internal model control using neural networks: application to processes with delay and design issues," *IEEE Transactions on Neural Networks*, vol. 11, no. 1, pp. 80 – 90, Jan 2000.

[28] W. F. Xie and A. B. Rad, "Fuzzy adaptive internal model control." *IEEE Trans. On Industrial Electronics*, vol. 47, no. 1, pp. 193–202, 2000.

[29] A. Fink and O. Nelles, "Nonlinear internal model control based on local linear neural networks," in *Proc. IEEE Systems, Man and Cybernetics Conf.*, vol. 1, Oct 2001, pp. 117–122.

[30] A. S. Bel-Hadj, A. El-Abed-Abdelkrim, and M. Benrejeb, "An internal model control strategy using artificial neural networks for a class of nonlinear systems," in *IEEE Int. Conf. on Systems, Man & Cybernetics*, vol. 5, 2002, p. 4.

[31] M. A. DENAI and S. A. ATTIA, "Fuzzy and neural control of an induction motor," *Int. J. Appl. Math. Comput. Sci.*, vol. 12, no. 2, p. 221233, 2002.

[32] G. Baoming, W. Xiangheng, S. Pengsheng, and J. Jingping, "Nonlinear internal-model control for switched reluctance drives," *IEEE TRANSACTIONS ON POWER ELECTRONICS*, vol. 17, no. 3, pp. 379–388, MAY 2002.

[33] M. Jalili-Kharaajoo, A. Rahmati, and F. Rashidi, "Internal model control based on locally linear model tree (lolimot) model with application to a ph neutralization process," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, Oct 2003, pp. 3051 – 3055.

[34] X. Wen, J. Zhang, Z. Zhao, and L. Liu, "Multi-model neural network imc," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, vol. 6, Aug 2004, pp. 3370 – 3374.

[35] G. Baoming, Z. Hongtao, Y. Ling, and J. JingPing, "Robust control methods for pm synchronous motor," in *Fifth World Congress on Intelligent Control and Automation*, vol. 5, June 2004, pp. 4437–4440.

[36] S. Su, J. Bao, and P. Lee, "Passivity based imc control for multivariable nonlinear systems," in *5th Asian Control Conference*, vol. 1, July 2004, pp. 135– 140.

[37] J. Shi and W. S. Lee, "Set point response and disturbance rejection tradeoff for second-order plus dead time processes," in *5th Asian Control Conference*, vol. 2, July 2004, pp. 881– 887.

[38] R. Suzuki, D. Yamashita, K. Mitani, and N. Kobayashi, "Internal model control for sumi-ink rubbing machine," in *SICE Annual Conference*, vol. 2, Aug 2004, pp. 1711– 1714.

[39] Q. Jia, T. Lee, and K. Liu, "Adaptive imc design for hdd servo control," in *Digests of the IEEE International Magnetics Conference*, San Diego, CA, April 2005, pp. 1691 –1692.

[40] J. Schwartz, D. Rivera, and K. Kempf, "Towards control-relevant forecasting in supply chain management," in *American Control Conference*, vol. 1, June 2005, pp. 202– 207.

[41] C. Yu, J. Zhu, and Z. Sun, "Nonlinear adaptive internal model control using neural networks for tilt rotor aircraft platform," in *IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications*, Helsinki University of Technology, Espoo, Finland, June 2005, pp. 535–536.

[42] J. Alvarez and J. Alvarez, "A tracking and regulation control scheme for discrete time nonlinear systems," in *In Proc. of the 28th conference on Decision and Control*, vol. 7, Oct 1989, pp. 535–536.

[43] K. Fruzzetti, A. Palazoglu, and K. MacDonald, "Non-linear model predictive control using hammerstein models," *J.Proc. Cont.*, vol. 7, no. 1, pp. 31–41, 1997.

[44] A. Banerjee and Y. Arkun, "Model predictive control of plant transitions using a new identification technique for interpolating non-linear models," *J.Proc. Cont.*, vol. 8, no. 5, pp. 441–457, 1998.

[45] Z. Ma, A. Jutan, and V. B. Bajic, "Nonlinear self-tuning controller for hammerstein plants with application to a pressure tank," *Int. J. of Computers, Systems and Signals*, vol. 1, no. 2, pp. 221–230, 2000.

[46] B. Bequette, "Practical approaches to nonlinear control: A review of process applications," *Nonlinear Model-based Process Control, NATO ASI Series*, vol. 353, pp. 3–32, 1998.

[47] P. V. Kokotovic and M. Arcak, "Constructive nonlinear control: A historical perspective," *Automatica*, vol. 37, pp. 637–662, 2001.

[48] P. Kokotovic, "Recent advances in nonlinear control," in *First International Symposium on Control, Communications and Signal Processing*, 2004, p. 721.

[49] W. S. McCulloch and W. Pitts, "A logical calculus of ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.

[50] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* N.Y.: Spartan, 1962.

[51] D. Rumelhart, G. Hinton, and R. Williams, *Learning Internal Representations by Error Propagation in Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, D. Rumelhart and J. McClelland, Eds. Cambridge, MA: MIT Press, 1986, vol. 1.

[52] ——, "Learning representations by back-propagation errors," *Nature*, no. 323, pp. 533–536, 1986.

[53] K. Funahashi, "On the approximate realization of continuous mappings by neural-networks," *Neural Networks*, vol. 2, pp. 183–192, 1989.

[54] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward network are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[55] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks,," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[56] J. Tanomaru and S. Omatu, "On the application of neural networks to control and inverted pendulum: an overview," in *Proceedings of the 30th SICE Annual Conference*, 1991, pp. 1151–1154.

[57] M. Greene and H. Tan, "Indirect adaptive control of a two-link robot arm using regularization neural networks," in *Proceedings of the International Conference on Industrial Electronics, Control and Instrumentation*, vol. 2, 1991, pp. 952–956.

[58] G. Widrow, B.; Plett, "Nonlinear adaptive inverse control," in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 2, Dec 1997, pp. 1032–1037.

[59] M. Norgaard, O. Ravn, N. K. Poulsen, and L. K. Hansen, *Neural Networks for Modelling and Contol of Dynamic Systems, A Practitioner's Handbook*. London: Springer-Verlag, 2000.

[60] S. Haykin, *Adaptive Filter Theroy*, 4th ed. NJ: Prentice Hall, Inc., 2002.

[61] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, 2nd ed. Netherlands: Kluwer Academic Publishers, 2002.

[62] N. Benvenuto and G. Cherubini, *Algorithms for Communication Systems and Their Applications*. England: John Wiley & Sons Ltd., 2004.

[63] M. M. Gupta, L. Jin, and N. Homma, *Static And Dynamic Neural Networks*. Hoboken, NJ: John Wiley and Sons, Inc., 2003.

[64] M. Morari and E. Zafiriou, *Robust Process Control*. NJ: Prentice Hall, 1989.

[65] C. A. Desoer and M. Vidyasagar, *Feedback Systems: Input-Output Properties*. NY: NY: Academic,, 1975.

[66] V. H. Nascimento and A. H. Sayed, "Unbiased and stable leakage-based adaptive filters," *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 47, no. 12, pp. 3261–3276, DECEMBER 1999.

[67] K. D. Young, "Design of variable structure model following control system," *IEEE Trans. Automat. Contr*, vol. AC-23, pp. 1079–1084, Dec. 1978.

[68] A. Brickwedde, "Microprocessor-based adaptive speed and position control for electric drives," *IEEE Trans. Ind. Applicat.*, vol. 21, pp. 1154–1161, Sept./Oct. 1985.

[69] H. Naitoh and S. Tadakuma, "Microprocessor based adjustable speed dc motor drives using model reference adaptive control," *IEEE Trans. Ind. Applicat.*, vol. 23, pp. 313–318, Mar./Apr. 1987.

[70] M. A. El-Sharkawi and C. H. Huang, "Variable structure tracking of dc motor for high performance applications," *IEEE Trans. Energy Conversion*, vol. 4, pp. 643–650, Dec. 1989.

[71] S. Weerasooriya and M. A. El-Sharkawi, "Identification and control of a dc motor using back-propagation neural networks," *IEEE Trans. Energy Conversion*, vol. 6, pp. 663–669, Dec. 1991.

[72] A. Rubaai and M. D. Kankam, "Adaptive real-time tracking controller for high performance induction motor drives using neural designs," in *Conf. Rec. IEEE/IAS Annu. Meeting*, vol. 3, San Diego, CA, Oct 1996, pp. 1709–1717.

[73] A. Rubaai and R. Kotaru, "Online identification and control of a dc motor using learning adaptation of neural networks," *IEEE Trans. On Industry Applications*, vol. 36, no. 3, pp. 935–942, May/June 2000.

[74] Q. M. Zhu, K. Warwick, and J. L. Douce, "Adaptive general predictive controller for nonlinear systems," *IEE Proc. Control Theory Applic.*, vol. 138, no. 1, pp. 33–40, 1991.

[75] J. G. Taylor and S. Commbes, "Learning higher-order correlations," *Neural Networks*, vol. 6, no. 3, pp. 423–428, 1993.

[76] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design.* New York: John Wiley & Sons, 1995.

LIST OF PUBLICATIONS

- M. Shafiq and Naveed R. Butt, "U-Model based adaptive IMC for nonlinear dynamic plants", in *Proceedings of the 10th IEEE International Conference on Emerging Technologies & Factory Automation*,Catania, Italy, vol. 1, 19-22 September 2005, pp. 955-959.

- Naveed R. Butt, M. Shafiq and T. Khan, "An adaptive root-solving controller for tracking of nonlinear dynamic plants", in: *Proceeding of the 2005 IEEE International Conference on Industrial Electronics and Control Applications*,Swissotel, Ecuador, 29-30 November 2005, pp. 144-149.

- Naveed R. Butt and M. Shafiq, "On the adaptive tracking of nonlinear dynamic plants using U-Model", Submitted to: *IMECHE Journal of Systems and Control Engineering*.

- Naveed R. Butt and M. Shafiq, "Higher-order neural network based root-solving controller for adaptive tracking of stable nonlinear plants", Presented at: *IEEE International Conference on Engineering of Intelligent Systems*, 22-23 April 2006, Islamabad, Pakistan.

VITAE

Naveed Razzaq Butt was born on the 5th of September, 1979, in Lahore, Pakistan. He earned his BS degree in Lasers & Opto-electronics from the GIK Institute (Pakistan) in 2002. He worked as a controls engineer on the *Riyadh Water Transmission System Line A & B SCADA Upgradation Project*, SWCC, Jubail. He joined KFUPM as a research assistant in 2004 and obtained a Master's degree in Systems Engineering (Automation & Control) in 2006.

Naveed Razzaq Butt has been actively involved in research from the very beginning of his career. His areas of interest include:

- Model Based Control systems
- System Identification
- Embedded and Intelligent systems
- Digital Signal Processing
- Communication Systems