Development of an Adaptive Neurofuzzy Controller

BY

LO CHANG HOW (B.ENG.)
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE
2003

Acknowledgments

Here's a salute to all who made this thesis possible.

First to God :

I would like to thank God for his grace. Without him providing the light, I would probably have not made it.

My family :

I would like to give appreciation to my parents, who have silently played a huge role in helping me through this tough period of time. My wife, Ting Ting, deserves special mention for all the sacrifices she had made for this piece of work to be possible.

My supervisor :

Special thanks goes to my supervisor, Dr. Tan Woei Wan. She had guided and helped me in many ways to make this thesis a success. Her patience with me is unparalleled.

My friends :

I would like to thank all my friends, who are always there to support and encourage me towards the end of this thesis. Special thanks goes out to the Reginald, Yongtian, Yuqiang, and Siva for companionship. My colleagues in Advance Control Technology Laboratory have also provided me much help, especially Yongsheng for his insights into control theory, and Vathi for the preparation of chemicals.

And to NUS :

Much appreciation goes to NUS for the research scholarship and facilities.

Chang How

July 2003

Contents

List of Figures

List of Tables

Summary

The "intelligence" of controllers may be improved by embedding *a prior* information about the process into the control scheme. One such intelligent control scheme utilizes a neurofuzzy controller as the feedforward controller. The data that is used to train the neurofuzzy controller on-line is obtained by adding the feedback error to the control action, in a method known as the feedback error learning strategy. Practical systems have successfully been controlled by the feedback learning algorithm. This thesis aims at improving the performance of such controllers by including the feedback error and its history in the learning rule. Emphasis is placed on developing a stability criteria and studying an alternative method for commissioning the adaptive controller. Analysis of the performance of the adaptive neurofuzzy controller is also extended to non-linear plants, with a liquid level plant and a pH neutralization process being used as test beds.

First, a stability guide for the neurofuzzy control scheme that is controlling a linear time invariant plants is established through insights gained from examining the stability of the learning algorithms individually. Simulation results verifying the feasibility of the stability criteria are presented.

Moving on to analyzing nonlinear plants, a comparison of the various feedback error learning strategies is performed by using a liquid level plant as the test bed. The study shows that the proposed feedback error learning rule strategy is be better suited for this control problem. Simulation results indicate that the proposed strategy's performance is superior to the other learning strategies, while experimental results demonstrate the feasibility of the proposed strategy in real world conditions.

As much as the incorporation of *a prior* information about the process may bring about more "intelligent" controllers, there is the associated difficulty in ascertaining the information's accuracy when the process dynamics changes drastically. The pH neutralization process, with its severe nonlinearity and sensitivity, is used to test whether there is merit in including structural information into the control scheme. Although the control task may be simplified by the inclusion of structural information, the controller has difficulties coping with changes to the buffering conditions. Even when the structural information is adapted on-line, simulation results show that the neurofuzzy control scheme is able to cope best without using the structural information. Also, the feasibility of using the neurofuzzy control scheme to handle an actual pH process is verified experimentally.

Chapter 1

Introduction

1.1 Adaptive Neurofuzzy Control

The never ending quest to improve the performance of control systems has led to the establishment of several major fields of research since the start of the modern control era. One such field is intelligent control, where the original inspiration came from either from nature or the human being. Within this field, fuzzy logic and neural networks are two popular research directions because they possess the universal approximation capability (Wang, 1992).

Fuzzy logic control has its roots in mimicking the reasoning capabilities of human beings. Through the incorporation of existing operator knowledge into a linguistic rule base, automated control of complex plants that have traditionally proved difficult to model can be achieved. However, the performance of these early fuzzy controllers depends entirely upon the initial design, and it is difficult to cope with unexpected changes in operating conditions or improve upon the existing controller's performance. This handicap is especially crippling in today's cutthroat industries, for process control is an important competitive advantage that one can have over its competitors.

Numerous methods, from training fuzzy logic controllers using conventional adaptive control approaches (Wang, 1994) to fuzzy relational modelling, have been used to identify the parameters of a fuzzy logic model (Czogala and Pedrycz, 1981).

One approach for adapting a fuzzy logic based controller is established when it was shown that a B-spline neural network is equivalent to a fuzzy network structure (Brown and Harris, 1994). This paves the way for fuzzy logic networks to be trained by neural network training algorithms. Unlike fuzzy logic, neural networks, which imitate the massive parallel structure of the human brain, usually treat the system to be modelled as a "black box", and train its adjustable parameters to minimize some performance criterion. Although good performance can be obtained, it is often difficult to obtain meaningful insights about the network. This problem can be alleviated by combining the linguistic reasoning of fuzzy systems with the learning abilities of neural networks by leveraging on the established equivalence relationship to form neurofuzzy networks. The combination of the two research directions of emulating the power of human beings is important, as one way to improve upon existing controllers is to make them more "intelligent" through the ability to embed more *a prior* knowledge into the controller, which in turn results in better control performance.

1.2 The Feedback Error Learning Strategy

To equip the neurofuzzy control scheme with learning capabilities, this thesis explores the usage of an interesting learning control scheme developed for robot manipulators (Kawato *et al.*, 1987). This control scheme, shown in Figure 1.1, is generally known as the Feedback Error Learning Strategy(FELS).

The learning control system consists of two parts- a feedforward controller, F, and a feedback controller, C. The aim of the feedforward controller is to compensate for the system dynamics in order to obtain good tracking accuracy. Assuming that the plant is stable, the feedforward controller having been trained to model the inverse plant dynamics in an ideal situation, or $F = P^{-1}$, will drive the output of the plant y to be equal to the reference r.

In the real world, the system will always be subjected to disturbances. The role of the feedback controller in the control scheme is to stabilize and minimize the deteriorative effects of the such stochastic or random disturbances. It also

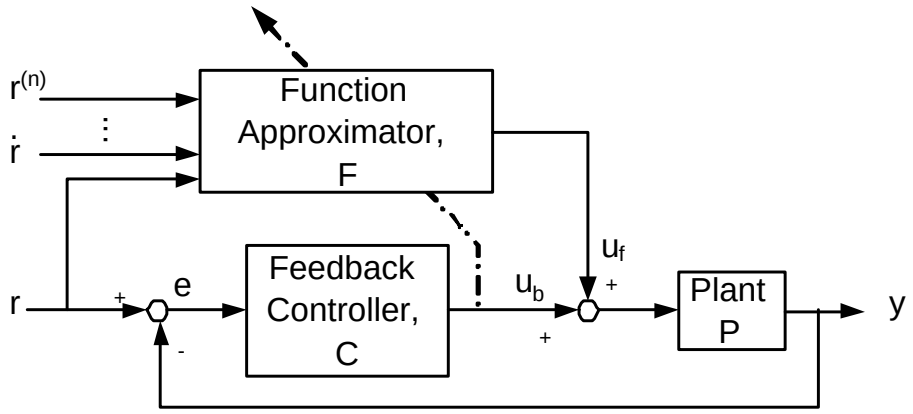

Figure 1.1. Feedback Error Learning Control Scheme

determines the minimum tracking performance at the beginning of the learning process as the feedforward controller at the start of the learning process is unlikely to have good performance when untrained.

Many methods have been proposed to enable the feedforward controller to learn the inverse plant dynamics. In general, they can be divided into indirect and direct estimation methods. In indirect estimation methods like adaptive inverse control (Widrow and Walach, 1996), a model of the plant is estimated before inverting the stable part to obtain the feedforward controller; whereas direct estimation methods do without the model in the estimation of the inverse model. Instead of designing a feedforward controller on the basis of a model, Kawato *et al.* (1987) proposed and implemented the feedforward controller as a function approximator. During control, the input-output relationship of the function approximator is adapted in such a way that it learns the inverse plant with the reproducible disturbances directly. The main difficulty lies in the selection of a learning signal that indicates how the input-output relationship should be adapted. Mimicking the way the neurons in our brain obtained the learning signal, Kawato *et al.* (1987) demonstrated that when the output of the feedback controller is used as a learning signal as in Figure 1.1, the function approximator is able to learn the inverse plant with reproducible disturbances.

This control scheme has been applied to a number of applications, such as

an automatic braking system for automobiles(Ohno *et al.*, 1994), camera system (Bruske *et al.*, 1997), robot manipulators (Kim *et al.*, 1996) and welding (Tzafestas *et al.*, 1997). The applications showed that the control scheme considerably improved upon the performance of the feedback controller and that it was able to obtain a good tracking performance without extensive modelling. When the FELS is compared to conventional adaptive control (Kraft and Campagna, 1990; Kim *et al.*, 1996; Tzafestas *et al.*, 1997), similar tracking performance can be expected from both schemes when an accurate plant model is made available for the latter. However, adaptive control is preferred in this instance as it converges comparatively faster. The tables are turned when an accurate plant model is unavailable, as the adaptive controller fails to obtain satisfactory tracking performance, unlike FELS. This conclusion demonstrates the usefulness of the FELS in real world situations, where accurate plant models are often difficult to obtain.

However, there are a few shortcomings in the function approximator that is used in the original formulation- the Multi-Layer Perceptron (MLP). Training of the MLP is often very slow due to the ill-conditioned performance surface imposed by the usage of the sigmoid function (Haykin, 1999). This is especially so when the data used to train the MLP is highly correlated, which inevitably occur in control problems. Moreover, the weights of the MLP may get trapped in local minima and fail to converge, as the trained weights are dependent on their initial values. Therefore, it may be necessary to perform several training experiments with different initial weights to obtain acceptable performance.

Improvements had been made to improve on its performance by incorpating the output error into the MLP (Gomi and Kawato, 1993), as well as the usage of multiple feedforward controllers to learn different tasks (Jacobs and Jordan, 1993). Nevertheless, the real difficulties with FELS lie with the usage of the MLP network. The obvious approach is to replace the MLP network with other function approximators. Kraft and Campagna (1990) replaced the MLP network with a Cerebellar Model Articulation Controller (CMAC) network that employ local basis functions. Experimental results showed that superior learning behaviour and

more accurate tracking performance could be attained. Recently, Velthuis and de Vrie (2000) used a B-spline network to control a Linear Motor Motion System. This decision is due to the relative ease in the choice of the distribution of the basis functions of a B-spline network over a CMAC network. However, the ability to embed information into the controller structure is not exploited.

1.3 Motivation of work

Inspired by the success of FELS, the notion of using the feedback error to identify the required desired control action, which is in turn used to update the weights of a neurofuzzy model online to represent the inverse plant dynamics was proposed (Tan, 1997). One advantage of this neurofuzzy control scheme is that it enables *a prior* information about the plant to be incorporated into the controller structure. The usually difficult task of choosing adequate parameters in adaptive control schemes is eased by relying on an approximate relationship with a conventional PI / PID controller. The self-learning neurofuzzy control scheme had been successfully employed to regulate the temperature in a liquid helium cryostat (Tan and Dexter, 1999).

While the feedback error learning strategy in the control scheme is able to perform reasonably well in some cases, the learning strategy is unable to cope when the rate of change of the control error is large (Tan and Lo, 2001*a*). This limitation led to modifications that included the derivatives of the feedback error into the learning strategy (Brandizzi *et al.*, 1999; Santos *et al.*, 2000). The on-line learning strategy was further refined in order to remove a restrictive assumption, and superior results were obtained when used to control linear time invariant plants (Lo, 2001).

Motivated by the success, this thesis aims to further explore the properties of the control scheme by studying the criteria needed for its stability, as well as looking at an alternative derivation of the commissioning strategy. The thesis also seeks to extend the analysis of the performance of the neurofuzzy control scheme to nonlinear plants, with a liquid level plant and a pH neutralization process being

used as test beds. Experimental verification to test the feasibility of the neurofuzzy control scheme on both plants are also carried out.

1.4 Organization of thesis

Chapter 2 presents the details of the neurofuzzy control scheme that is evaluated in this thesis. First, the notion of inverse learning, which is the main idea behind the control scheme, is described. A description of the neurofuzzy model and its modelling capability is presented next. Details of the control structure are then shown, and the role of each component in the control scheme described. The original on-line learning mechanism follows next, together with the modifications that had been suggested to improve the control scheme's performance. A new derivation of the commissioning strategy for the proposed feedback error learning strategy is also presented.

Development of the neurofuzzy control scheme is made in Chapter 3 by deriving stability conditions. Through considering the stability of each part of the learning process individually, insights into the operation of the control scheme were made. Based on the observations, conditions for maintaining the stability of the adaptive controller were derived. Simulations are then presented to verify the proposed stability criteria.

Next, the performance of the neurofuzzy control scheme is analyzed through a liquid level control problem. A comparison of the control performances of the various feedback error learning strategies is presented, and an alternative commissioning guide for the proposed feedback error learning strategy is evaluated. Experimental verification of the practicality of the proposed learning strategy with neurofuzzy control scheme on a liquid level plant is then documented.

Thus far, the neurofuzzy control scheme was evaluated using linear or mildly nonlinear plants. In Chapter 5, control of a highly nonlinear system, the pH neutralization process, is attempted. The pH plant is first introduced, and the process is shown to approximate a Wiener model. A study of the merits of incorporating *a prior* structural information into the neurofuzzy control scheme is then carried out.

The control scheme is tested on a pilot pH plant, and the experimental results show that the neurofuzzy control scheme can provide reasonable control performance.

Lastly, conclusions about the work in this thesis is described in Chapter 6, followed by suggestions about possible future work.

Chapter 2

The Neurofuzzy Control Scheme

2.1 Introduction

This chapter provides a review of the neurofuzzy control scheme that is evaluated in this thesis. Various properties that are used in the analysis and development of the control scheme in the later chapters of this thesis are described.

The organization of this chapter is as follows. First, the structure of the control scheme is presented, with a brief explanation of the role of each component in the control scheme. Section 2.5 continues with a description of the control scheme's on-line learning mechanism.

2.2 Inverse Learning

As mentioned in the previous chapter, the aim of the self-learning control scheme is to determine the parameters of the neurofuzzy feedforward controller such that it models the process's inverse input-output mapping. Suppose the plant can be expressed as a k^{th} order discrete non-linear series :

$$y(t) = P\{y(t-1), y(t-2), ..., y(t-k), u(t-t_d), u(t-t_d-1), u(t-t_d-k+1)\} \quad (2.1)$$

where t_d is equal to the plant delay expressed as a multiple of sampling instants plus one. The additional delay is the result of cascading the systems with a zero-order hold.

Assuming that a stable inverse plant model exists for the controlled system, the neurofuzzy controller should be trained to model the following components :

$$u(t-t_d) = Q\{y(t), y(t-1), ..., y(t-k), u(t-t_d-1), ..., u(t-t_d-k+1)\} \quad (2.2)$$

However, this model is not realizable as it is not causal. To resolve this problem, the inverse model is constructed by replacing the plant's output signal by the reference signal, with the expectation that through training, the plant's output will approach the reference trajectory. It is possible to know the reference signal t_d sampling instants ahead of time as the user of the system decides on the reference trajectory. Hence, the resulting control action by the neurofuzzy feedforward model is

$$u(t) = Q\{r(t+t_d), r(t+t_d-1), ..., r(t+t_d-k), u(t-1), u(t-2), u(t-k+1)\} \quad (2.3)$$

Next,the neurofuzzy model that is used to model Equation (2.3) is described.

2.3 The Neurofuzzy Model

The neurofuzzy model that is employed in this thesis is a B-spline network that uses basis functions for approximation purposes. B-spline networks have been employed as surface-fitting algorithms within the graphical visualization community for many years. The difference between classifying B-spline networks as a surface fitting algorithm and a neural network lies in the way in which the linear coefficients (weights) are generated. While the neural network adjusts its weights iteratively to reproduce a particular function, the off-line or batch B-spline algorithm typically generates the coefficients by matrix inversion or using conjugate gradient. The reason for the choice of this model structure is that it provides a direct link between neural networks and fuzzy logic systems, thus making the embedment of *a prior* information easier.

Figure 2.1 shows the structure of the neurofuzzy model. There are two parts to the network : a static, nonlinear, topology conserving map and an adaptive linear mapping.

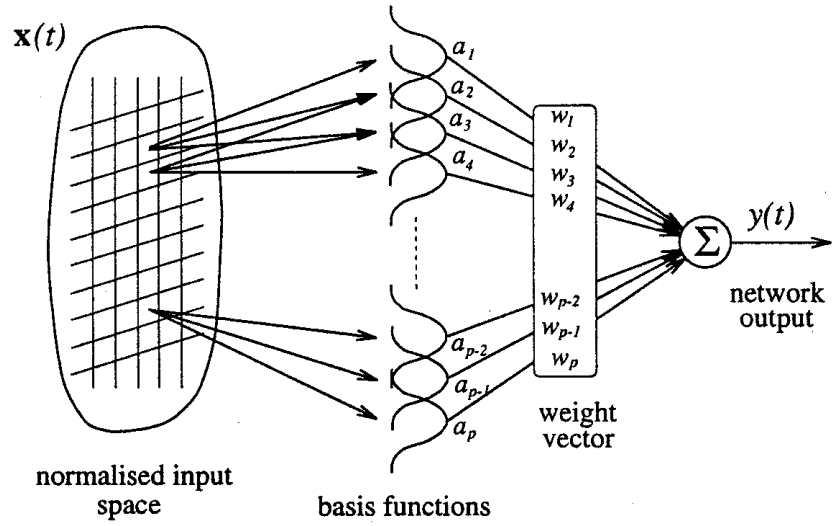

Figure 2.1. The neurofuzzy model

2.3.1 Nonlinear transformation by basis functions

The power of the B-spline network, or neurofuzzy system, in modelling non-linear functions comes from the non-linear transformation of the input vector $\mathbf{x}$ by the basis functions (or fuzzy sets) of the network. Suppose that for each input x_i, the input space is spanned by m_i basis functions. For a B-spline network, the stable recurrence relationship for evaluating the output of the j^{th} univariate B-spline basis function of order k is defined as (Cox, 1972):

$$\begin{aligned} N_k^j(x) &= \left(\frac{x-\chi_{j-k}}{\chi_{j-1}-\chi_{j-k}}\right) N_{k-1}^{j-1}(x) + \left(\frac{\chi_j - x}{\chi_j - \chi_{j-k+1}}\right) N_{k-1}^j(x) \\ N_1^j(x) &= \begin{cases} 1 & \text{if } x \epsilon I_j \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{2.4}$$

where χ_j is the j^{th} knot and $I_j = [\chi_{j-1}, \chi_j)$ is the j^{th} interval. The shapes of the univariate basis functions with orders 1 to 4 are depicted in Figure 2.2, with 11 basis functions of equal support spanning a normalized input space. From the recurrence relationship in Equation (2.4), it can be derived that the basis function is differentiable, up to $k-2$ order, and continuous up to $k-1$ order. At each interval, k B-spline weights are used to represent a polynomial of order k, which in turn determines the modelling capability and smoothness of the basis function

output.

The univariate B-spline basis functions can also be interpreted as fuzzy sets with singleton outputs (Brown and Harris, 1994). This property enables linguistic meaning to be assigned to a basis function as in a fuzzy set, and its output to be interpreted as the degree of truth in the meaning.

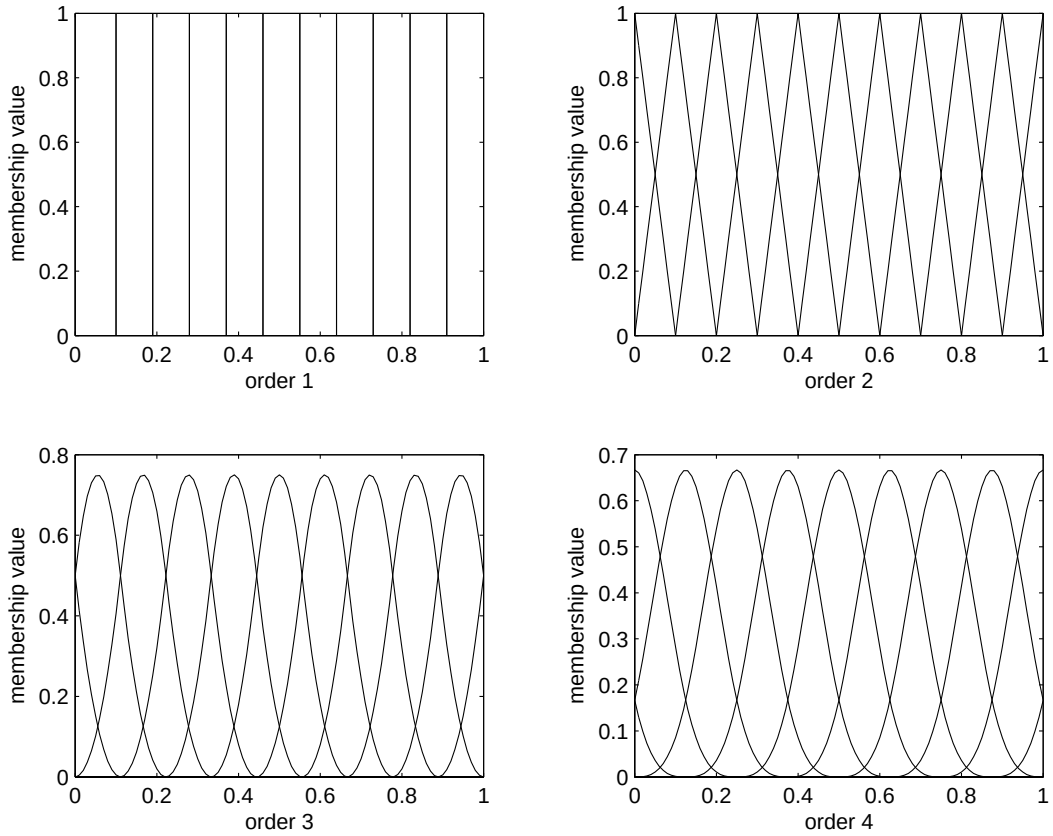

Figure 2.2. Univariate B-spline basis functions of orders 1-4

In addition, the B-spline basis functions that are generated by the recurrence relationship in Equation (2.4) have many desirable properties. Some important properties are : (i) the basis functions have a bounded support, and (ii) the output of the basis function is positive on its support, i.e.

$$\begin{aligned} N_k^j(x) &= 0, x \notin [\chi_{j-k}, \chi_j], \text{ and} \\ N_k^j(x) &> 0, x \in (\chi_{j-k}, \chi_j). \end{aligned} \tag{2.5}$$

The basis functions also form a partition of unity, meaning that the sum of the

outputs of the basis functions is always one, or

$$\sum_j N_k^j(x_i) \equiv 1, x \in [x^{min}, x^{max}] \tag{2.6}$$

Let the membership vector $\mu(x_i)$ generated from the m_i univariate B-spline functions for the x_i input be

$$\mu(x_i) = \begin{bmatrix} N_k^1(x_i) \\ N_k^2(x_i) \\ \vdots \\ N_k^{m_i}(x_i) \end{bmatrix} \tag{2.7}$$

The extension of the univariate basis functions to form the multivariate basis functions is achieved via the usage of the Kronecker tensor product to combine all the n membership vectors as follows :

$$a(\mathbf{x}) = \prod_{i=1}^{n} \mu(x_i) \tag{2.8}$$

As one and only one univariate basis from each input is used for each multivariate basis function, all the desirable properties of the univariate B-spline basis functions are extended in a natural way to the multivariate basis functions (Brown and Harris, 1994). For example, the order of the univariate basis functions used determines the smoothness of the multivariate basis functions. The equivalence to a fuzzy logic model is the usage of a complete set of rules, and the s and t norms in the fuzzy composition process to form the fuzzy output distribution. Viewed in this context, this step allows for the model to produce *sensible* outputs for previously unseen inputs, and is equivalent to generalization in neural networks, or interpolation and local extrapolation in approximation theory (Wang, 1997).

2.3.2 Adaptive Linear Mapping

The last step is to generate the output of the network by multiplying the output of the multivariate basis functions with their associated weights. It has the same

form as using the center of gravity defuzzification method in fuzzy logic :

$$\begin{aligned} u_f &= \sum_{i=1}^{p} a_i w_i \\ &= \mathbf{a}^T \mathbf{w} \end{aligned} \tag{2.9}$$

Having described the operation of the neurofuzzy model, the question about its modelling capability will be addressed.

2.3.3 Modelling capability of the neurofuzzy model

For illustration purposes, a 2 (x_1, x_2) input network with 2^{nd} order regularly spaced (triangular) basis functions is used to demonstrate the modelling capability of the neurofuzzy model. Suppose the inputs x_1 and x_2 lie between the intervals $[\chi_{j,1}, \chi_{j+1,1}]$ and $[\chi_{k,2}, \chi_{k+1,2}]$. According to Equation (2.9), the output of the network is

$$\begin{aligned} \mathbf{a}^T \mathbf{w} &= \frac{1}{(\chi_{j+1,1} - \chi_{j,1})(\chi_{k,2} - \chi_{k+1,2})} \\ &\begin{bmatrix} \chi_{j+1,1}\chi_{k,2} & -\chi_{k,2}x_1 & -\chi_{j+1,1}x_2 & +x_1x_2 \\ \chi_{j+1,1}\chi_{k+1,2} & +\chi_{k+1,2}x_1 & +\chi_{j+1,1}x_2 & -x_1x_2 \\ -\chi_{j,1}\chi_{k,2} & +\chi_{k,2}x_1 & +\chi_{j,1}x_2 & -x_1x_2 \\ -\chi_{j,1}\chi_{k+1,2} & -\chi_{k+1,2}x_1 & -\chi_{j,1}x_2 & +x_1x_2 \end{bmatrix}^T \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \end{aligned} \tag{2.10}$$

In order to simplify the above expression, it is assumed that the input space of the interval considered is normalized, i.e., $\chi_{j+1,1} = \chi_{k+1,2} = 1$ and $\chi_{j,1} = \chi_{k,2} = 0$.

Then, Equation (2.10) becomes

$$
\begin{aligned}
\mathbf{a}^T\mathbf{w} &= \begin{bmatrix} 1 & -x_1 & -x_2 & +x_1x_2 \\ & & x_2 & -x_1x_2 \\ & x_1 & & -x_1x_2 \\ & & & x_1x_2 \end{bmatrix}^T \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \\
&= \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1x_2 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \\
&= \mathbf{x}^T\mathbf{H}\mathbf{w} \\
&= (1 - x_1 - x_2 + x_1x_2)\, w_1 + (x_2 - x_1x_2)\, w_2 + (x_1 - x_1x_2)\, w_3 + x_1x_2w_4 \\
&= w_1 + x_1\,(w_3 - w_1) + x_2\,(w_2 - w_1) + x_1x_2\,(w_1 - w_2 - w_3 + w_4) \\
&= \theta_1 + \theta_2x_1 + \theta_3x_2 + \theta_4x_1x_2 \qquad (2.11)
\end{aligned}
$$

where $\theta_1 = w_1$, $\theta_2 = w_3 - w_1$, $\theta_3 = w_2 - w_1$, and $\theta_4 = w_1 - w_2 - w_3 + w_4$. Since $\mathbf{H}$ is not singular, the arbitrary θ_i values can be constructed from w_i, and thus the neurofuzzy controller has the ability to model the polynomial function $f(1, x_1, x_2, x_1x_2)$ for the interval investigated.

Using the property that the basis functions are local in nature, similar conclusions on the type of polynomial fit across the entire input range can be made. However, the ability to choose arbitrary values for all θ are lost when the order of the basis functions used are more than 1 (or piecewise constant), as each basis function will span across more than 1 knot (or apex). This is a tradeoff for improving the smoothness of the network's output.

The magnitude of each linearly transformed weight w_i shows the importance of the term in the modelling process. Those terms, whose w_i are relatively small after training, are probably not important, and hence may be pruned off (as in neural networks) to improve the robustness of the model.

One general criticism of this network is that the number of weight vector increases exponentially with the number of inputs. This is due to the assignment

of one weight for each permutation of all the inputs to the order of the B-spline network, which in turn lays the modelling power of the network. The following section shall present the structure of the neurofuzzy control scheme, and the roles of the various components.

2.4 Structure of the Neurofuzzy Control Scheme

Figure 2.3 shows the block diagram of the self-learning neurofuzzy controller that utilizes the feedback error learning strategy to perform on-line training (Tan, 1997). There are four main components in the control scheme : (i) a feedforward controller, (ii) an on-line identification mechanism, (iii) a proportional controller, and (iv) a reference model.

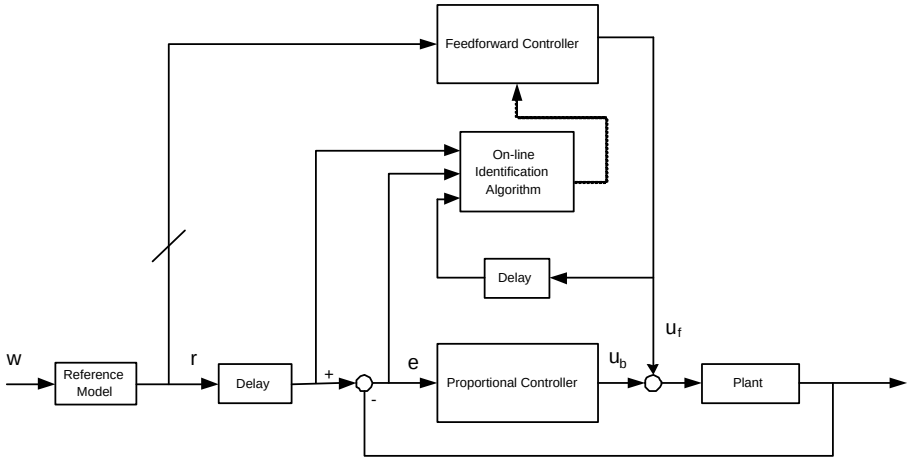

Figure 2.3. General structure of the neurofuzzy control scheme

The role of the feedforward controller is to model the inverse plant dynamics through the on-line identification mechanism, and is the crux of attaining good control performance. Although "perfect" control is theoretically attainable, an exact inverse model is difficult, if not impossible, to derive in practice, and therefore, the feedforward controller will exhibit finite modelling errors. Hence, a proportional controller is included in the feedback path to compensate for modelling mismatches. The proportional controller is expected to act as a stabilizer, especially at the start

of the learning process when the neurofuzzy feedforward controller is unlikely to exhibit good control.

Another essential component of the scheme is the reference model. It filters the step changes in the set points in order to provide a reference trajectory that may be followed by the plant given the physical constraints and plant dynamics. The output error, $e(t)$, used in both the feedback controller and the FELS for the tuning of the feedforward controller is generated by :

$$e(t) = r(t) - y(t) \tag{2.12}$$

By using the reference model to manipulate the reference signal $r(t)$, the rate at which the output error changes may be dictated by the designer of the control scheme, thus presenting an additional degree of freedom in tweaking the error to control the response.

Having presented the framework of the control scheme and the neurofuzzy model described, the next section describes the on-line learning mechanism used to train the neurofuzzy model.

2.5 The On-line Learning Mechanism

The on-line learning mechanism consists of two parts, namely an estimation algorithm for the required control action and an update algorithm to store the estimated required control action into the neurofuzzy model.

2.5.1 Estimating the required control action

The feedback error learning strategy (Kawato *et al.*, 1987) is based on the observation that a nonzero feedback error is caused by an incorrect feedforward control action. When there are no unmeasurable disturbances and the feedforward controller drives the plant with the appropriate control action, the feedback error $e(t)$, will be zero. For linear systems at steady state, the output feedback error is proportional to the error in the control action supplied. Consequently, the feedback

error may be viewed as a modelling error and may be used as a corrective term in the estimation of the required control action. Thus, an estimate of the control action needed, $\hat{U}(t)$, can be generated as (Tan, 1997):

$$\hat{U}(t) = u_f(t - t_d) + \gamma e(t) \tag{2.13}$$

where γ is the on-line learning rate. The reason for using the output of the feedforward controller t_d sampling instants ago, $u_f(t - t_d)$, to estimate the desired control action is because the inherent plant delay causes the effects of the control action administered at time t to show up t_d samples later. This means that the feedback error, $e(t)$ is due to the control error occurring at the instant $t - t_d$. Therefore, it makes sense for the desired control action, $\hat{U}(t)$, to be a linear combination of the two signals. Thus, the feedback error learning strategy can be viewed as a iterative method that searches for the desired control action.

2.5.2 Storing the estimated desired control action

The estimated control action that will enable the plant output to track the reference signal is updated into the memory of the neurofuzzy controller via any recursive identification algorithm. Since the neurofuzzy model is linear-in-the-parameters, the Normalized Least Mean Squares (NLMS) algorithm was chosen for its low computational requirements :

$$\mathbf{w}(t) = \mathbf{w}(t-1) + \frac{\delta \mathbf{a}(t)}{\mathbf{a}^T(t)\mathbf{a}(t)}\epsilon(t) \tag{2.14}$$

where $\epsilon = \hat{U}(t) - a^T(t)w(t-1)$ is the error in the control action space. The usage of NLMS in the identification algorithm is desirable as it is able to minimize the *posterior* error and has minimal disturbance effect upon the weights (Brown and Harris, 1994).

2.5.3 Approximate Relationship between control scheme and a PI Controller

When the two optimization algorithms (Equation (2.14) and Equation (2.13)) are combined together, the feedforward control action generated by the neurofuzzy model can be expressed as

$$\begin{aligned} u_f(t) &= \mathbf{a}(t)\mathbf{w}(t) \\ &= \mathbf{a}(t)\left(\mathbf{w}(t-1) + \frac{\delta\mathbf{a}(t)}{\mathbf{a}^T(t)\mathbf{a}(t)}\left(\hat{U}(t) - \mathbf{a}^T(t)\mathbf{w}(t-1)\right)\right) \\ &= \mathbf{a}(t)\left(\mathbf{w}(t-1) + \frac{\delta\mathbf{a}(t)}{\mathbf{a}^T(t)\mathbf{a}(t)}\left(u_f(t-t_d) + \gamma e(t)) - \mathbf{a}^T(t)\mathbf{w}(t-1)\right)\right) \end{aligned} \tag{2.15}$$

Assuming that the transformed input vector , $\mathbf{a}$, is independent of time, Equation (2.15) becomes

$$u_f(t) = \delta u_f(t-t_d) + (1-\delta)u_f(t-1) + \delta\gamma e(t) \tag{2.16}$$

The total control action received by the plant is

$$U(t) = u_f(t) + k_p e(t) \tag{2.17}$$

Hence,

$$U(t) = \delta u_f(t-t_d) + (1-\delta)u_f(t-1) + \delta\gamma e(t) + k_p e(t) \tag{2.18}$$

Performing Z-transform on Equation (2.18) and rearranging it, a discrete transfer function relating the control action and the error is obtained as

$$\frac{U(z^{-1})}{E(z^{-1})} = k_p + \frac{\delta\gamma}{(1-z^{-1}) + \delta(z^{-1} - z^{-t_d})} \tag{2.19}$$

Since the sum of the geometric progression $z^{-1}, z^{-2}, z^{-3}, \ldots, z^{-t_d+1}$ is

$$z^{-1} + z^{-2} + \ldots + z^{-t_d+1} = \frac{z^{-1}(1-z^{-t_d+1})}{1-z^{-1}} \tag{2.20}$$

the total control action (Equation (2.19)) can be written as

$$\frac{U(z^{-1})}{E(z^{-1})} = k_p + H(z^{-1})\frac{\delta\gamma}{(1-z^{-1})} \tag{2.21}$$

where $H(z^{-1}) = \frac{1}{1+\delta(z^{-1}+z^{-2}+...+z^{-t_d+1})}$ is the transfer function of a low pass filter with a static gain of $\frac{1}{1+\delta(t_d-1)}$. The low frequency Z-transform of the control scheme can therefore be approximated by the following transfer function :

$$\frac{U(z^{-1})}{E(z^{-1})} = k_p + \frac{\delta\gamma}{(1-z^{-1})(1+\delta(t_d-1))} \tag{2.22}$$

Comparing Equation (2.22) with the discrete time implementation of a PI controller with gain K and integral time T_i of the form (Clarke, 1984) :

$$\frac{U(z^{-1})}{E(z^{-1})} = K\left(1 - \frac{h}{2T_i}\right) + \frac{Kh}{T_i(1-z^{-1})} \tag{2.23}$$

an approximate relationship is established between the two controllers (Tan and Dexter, 1999) :

$$k_p = K\left(1 - \frac{h}{2T_i}\right) \tag{2.24}$$

$$\frac{\delta\gamma}{1+\delta(t_d-1)} = \frac{Kh}{T_i} \tag{2.25}$$

with K and T_i being the proportional gain and integral time of the PI controller. k_p, δ, γ, and h are the proportional gain of the feedback controller, learning rate of the NLMS algorithm, FELS learning rate, and sampling period of the neurofuzzy control scheme.

The establishment of this approximate relationship enables the controller parameters to be chosen more easily. Moreover, the initial system performance is similar to the PI controlled system. The difference is that the self-learning control scheme will gradually improve upon its performance in an automated manner with time. Therefore, it is in a better position to cope with gradual changes to the plant with continuous learning, as with all adaptive systems.

Having described the original formulation of the learning mechanism as presented in (Tan, 1997), the next section looks at the modifications made to the mechanism to improve upon its convergence rate.

2.6 Improvements to the learning mechanism

2.6.1 The modified FELS

Several modifications have been proposed to improve upon the learning rate of the self-learning control scheme (Brandizzi *et al.*, 1999; Santos *et al.*, 2000; Lo, 2001). Specifically, improvements to the estimation of the control action may be achieved by adding the derivatives of the feedback error into Equation (2.13) up to the order of the plant being controlled (Brandizzi *et al.*, 1999; Santos *et al.*, 2000) :

$$\hat{U}(t) = u_f(t - t_d) + \gamma \left(e(t) + \sum_{i=1}^{n} \lambda_i e^i(t) \right) \tag{2.26}$$

where n is the order of the plant, e^i denotes the order of the derivative of the feedback error, and λ_i are user defined constants. The motivation for the additional term can be viewed as increasing the *a prior* knowledge implanted into the identification algorithm, and empirical results in (Lo, 2001) demonstrated an improvement in the convergence rate of the weights. Using the modified feedback error learning strategy, an approximate relationship relating the neurofuzzy control scheme to conventional PI/PID controllers can also be derived. Consider a first order plant of the form

$$G(s) = \frac{K_g e^{-s\tau_d}}{\tau s + 1} \tag{2.27}$$

where K_g, τ and τ_d are the static gain, time constant and deadtime of the process. An approximate relationship with a PI controller is established as (Brandizzi *et al.*, 1999)

$$k_p = K \left(1 - \frac{h + 2\lambda_1}{2T_i} \right) \tag{2.28a}$$

$$\frac{\gamma\delta}{1 + \delta(t_d - 1)} = \frac{Kh}{T_i} \tag{2.28b}$$

where K and T_i are the proportional gain and integral time of the equivalent PI controller, and t_d is the dead-time expressed as the number of sampling intervals. Similarly, Santos *et al.* (2000) established the conditions under which the self-learning neurofuzzy controller is equivalent to a PID controller for a second order

plant of the form

$$\tau_1\tau_2\ddot{y}(t) + (\tau_1 + \tau_2)\dot{y}(t) + y(t) = K_g u(\tau - \tau_d) \tag{2.29}$$

with K_g is the static gain, τ_1 and τ_2 are the time constants, τ_d is the dead-time, and $u(t)$ is the applied control action. For such a second order plant, the self-learning control scheme can be shown to be equivalent to a discrete-time PID control algorithm when the output of the reference model is close to steady state by the following set of equations (Santos and Dexter, 2001) :

$$k_p = K\left(1 - \frac{\lambda_1}{T_i}\right) \tag{2.30a}$$

$$\lambda_2 = T_i T_d \tag{2.30b}$$

$$\frac{\gamma\delta}{1 + \delta(t_d - 1)} = \frac{Kh}{T_i} \tag{2.30c}$$

where K, T_i and T_d are the proportional gain, the integral time, and the derivative action of the PID controller, while h is the sampling interval.

2.6.2 The proposed FELS

One problem with the modified learning strategies is that the plant must have a relatively long dead-time compared with its time constant for the proportional gain, k_p, to assume positive values when Equations (2.28) or (2.30) and Ziegler-Nichols tuning rules are used to commission the control scheme (Tan and Lo, 2001*a*; Lo and Tan, 2001*b*). Moreover, it is found through simulations that the λ_i chosen using this method may not give rise to a stable closed-loop system if the weights of the neurofuzzy controller are not initialized close to their desired values, as the rates of change of error will be large.

To alleviate these problems, Lo (2001) improved upon the estimation of the desired control action by taking into account the interaction between the conventional proportional controller and the neurofuzzy controller. Suppose the system to be controlled is the first order plant defined in Equation (2.27). The output of the plant, when controlled by the control scheme, is

$$\tau\dot{y}(t) + y(t) = K_g\hat{u_f}(t - t_d) + K_g k_p e(t - t_d) \tag{2.31}$$

When the feedforward controller has learnt the inverse plant dynamics exactly, the desired control action assumes the form :

$$\tau\dot{r}(t) + r(t) = K_g\hat{u_f}(t - t_d) \tag{2.32}$$

Subtracting Equation (2.32) from Equation (2.31) and rearranging,

$$\hat{u_f}(t - t_d) = u_f(t - t_d) + \gamma(e(t) + \lambda_1\dot{e}(t)) + k_p e(t - t_d) \tag{2.33}$$

where

$$\gamma = \frac{1}{K_g} \tag{2.34}$$

$$\lambda_1 = \tau \tag{2.35}$$

Equation (2.33) is the proposed FELS for controlling a first order plant. Extending Equation (2.33), the proposed strategy for a general n^{th} order plant is

$$\hat{U}(t) = u_f(t - t_d) + \gamma\left(e(t) + \sum_{i=1}^{n}\lambda_i e^i(t)\right) + k_p e(t - t_d) \tag{2.36}$$

The approximate relationship between the control scheme using the proposed strategy and a conventional linear controller is derived in a way similar to that described in Section 2.5.3. To simplify the derivation of the relationship, consider the first order plus dead-time plant (Equation (2.27)). Assuming that the transformed input vector **a** does not vary with time, the neurofuzzy model's output can be obtained by combining Equation (2.14) and Equation (2.33) to become

$$\begin{aligned} U(t) &= u_f(t - t_d) + k_p e(t) \\ &= (1 - \delta)u_f(t - 1) + k_p e(t) \\ &\quad +\delta\left(u_f(t - t_d) + \gamma(e(t) + \lambda_1\dot{e}(t)) + \delta k_p e(t - t_d)\right) \end{aligned} \tag{2.37}$$

Performing the Z-transform on Equation (2.37) results in

$$\frac{U(z^{-1})}{E(z^{-1})} = \frac{k_p\left(1 - (1 - \delta)(z^{-1} - z^{-t_d})\right) + \delta\gamma\left(1 + \lambda_1(1 - z^{-1})\right)}{(1 - z^{-1}) + \delta(z^{-1} - z^{-t_d})} \tag{2.38}$$

Using Equation (2.20), Equation (2.38) can be written as

$$\frac{U(z^{-1})}{E(z^{-1})} = H(z^{-1})\frac{k_p\left(1 - (1 - \delta)(z^{-1} - z^{-t_d})\right) + \delta\gamma\left(1 + \lambda_1(1 - z^{-1})\right)}{(1 - z^{-1})} \tag{2.39}$$

where $H(z^{-1}) = \frac{1}{1+\delta(z^{-1}+z^{-2}+...+z^{-t_d+1})}$ is the transfer function of a low pass filter with a static gain of $\frac{1}{1+\delta(t_d-1)}$. The low frequency Z-transform of the self-learning control scheme for first order plants can be approximated to be

$$\frac{U(z^{-1})}{E(z^{-1})} = \frac{k_p\,(1-(1-\delta)(z^{-1}-z^{-t_d})) + \delta\gamma\,(1+\lambda_1(1-z^{-1}))}{(1-z^{-1})(1+\delta(t_d-1))} \tag{2.40}$$

If δ is assumed to have unity value, Equation (2.40) will then take the form :

$$\frac{U(z^{-1})}{E(z^{-1})} = \frac{k_p + \gamma\,(1+\lambda_1(1-z^{-1}))}{t_d(1-z^{-1})} \tag{2.41}$$

The assumption that the update rate is unity for the NLMS algorithm implies that the algorithm updates the weights such that the weight vector is on the solution hyperplane. Rearranging Equation (2.41), the following expression is obtained :

$$\frac{U(z^{-1})}{E(z^{-1})} = \frac{\gamma\lambda_1}{t_d} + \frac{k_p+\gamma}{t_d(1-z^{-1})} \tag{2.42}$$

Comparing Equation (2.42) with a discrete PI controller (Equation (2.23)) results in the following relationship :

$$K = \frac{\gamma\lambda_1}{t_d} \tag{2.43a}$$

$$\frac{Kh}{T_i} = \frac{k_p+\gamma}{t_d} \tag{2.43b}$$

Equation (2.43) can be used as a starting point for commissioning the parameters of the self-learning controler used to regulate first order plants. Since there are 3 variables (λ_1, γ, k_p) to select and only two equations, there is an additional freedom of choice left in the commissioning strategy. Drawing inspiration from the derivation of the proposed FELS, Equation (2.34), re-presented as Equation (2.44), can be used to suggest parameter values for the control scheme :

$$\gamma = \frac{1}{K_g} \tag{2.44a}$$

$$\lambda_1 = \tau_1 \tag{2.44b}$$

As only 1 equation is needed, Equation (2.44a) is selected as the last commissioning equation because it is observed that large values of λ_1 may lead to

instability (Lo, 2001). Superior convergence rates were obtained using the proposed learning strategy when compared with both the modified and the original FELS when linear plants are controlled (Lo, 2001). Moreover, the proposed FELS does not require that the plant's dead-time must be long when compared with its time constant for the commissioning equations to yield a positive proportional gain, k_p. Since the above strategy is based on the intuition that a large λ_1 may give rise to stability problems, the alternative strategy of setting λ_1 to the plant time constant, τ (Equation (2.44b)) is investigated in this thesis. Using this relation, the neurofuzzy control scheme is related to a PI controller's parameters using the following equations :

$$\lambda_1 = \tau \tag{2.45a}$$

$$\gamma = \frac{Kt_d}{\lambda_1} \tag{2.45b}$$

$$k_p = \frac{Kht_d}{T_i} - \gamma \tag{2.45c}$$

This commissioning strategy will be compared against the generic form ($\gamma = \frac{1}{K_g}$) in the next chapter using a liquid level plant.

2.7 Conclusion

A review of the neurofuzzy control scheme that is used in this thesis is given. The main ideas behind the control scheme, and its operation is given. Improvements made to the on-line learning mechanism are also described.

Chapter 3

Stability Criterion for the Neurofuzzy Control Scheme

3.1 Introduction

Although guidelines for choosing the learning parameters in the control scheme have been proposed, the lack of a stability proof stands in the way of theoretical completeness. The main difficulty arises from the seemingly "ad hoc" usage of two optimization strategies in estimating the required control action and updating the weights of the neurofuzzy controller.

This chapter takes a journey through the motivation and proofs of stability of the individual update laws used in the control scheme. This allows for an insight into the limitations inherent in the control scheme before an attempt is made to derive the stability criterion for the self-learning control scheme.

3.2 Stability of Feedback Error Learning Strategy

3.2.1 Motivation of Inverse Control

The essence of the self-learning control scheme is to exploit the learning capabilities of the neurofuzzy controller so that it emulates the inverse process dynamics. Consider a discrete linear plant of the form

$$A_p y(t) = B_p U(t - t_d) \tag{3.1}$$

where $A_p = 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_n z^{-n}$

$B_p = b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_m z^{-m}$

t_d is the delay (in number of samples) of the process

If the control objective is for the plant to follow a reference trajectory, $r(t-t_d)$, the feedback error can be defined as

$$e(t) = r(t - t_d) - y(t) \tag{3.2}$$

When "perfect" control of the system is obtained, $e(t) = 0$, or

$$y(t) = r(t - t_d) \tag{3.3}$$

Substituting Equation (3.3) into Equation (3.1), the idea in Inverse Control is to invert the plant so that the ideal control action $U^*(t - t_d)$ is

$$B_p U^*(t - t_d) = A_p r(t - t_d) \tag{3.4}$$

The error dynamics of the closed loop system can then be obtained by substituting Equation (3.4) into Equation (3.1) to obtain

$$A_p e(t) = 0 \tag{3.5}$$

Thus, if the magnitude of the roots of A_p is less than 1, the output error will decay to zero. This equation brings to light an underlying limitation of Inverse

Control- the plant must be stable, or must be stabilized. Furthermore, the rate of decay of the output error, even in the knowledge of the ideal control action, depends entirely upon the original plant's dynamics if the initial error is nonzero.

Next, the two update laws used in the self-learning control scheme are analyzed independently of each other in order to establish a feel for the convergence requirements of each update law.

3.2.2 Convergence criterion for the Feedback Error Learning Strategy

The essence of the Feedback Error Learning Strategy (FELS) is to estimate the required control action by updating the control action with a portion of the feedback error and its history :

$$U(t) = U(t - t_d) + \gamma' e(t) \tag{3.6}$$

where $\gamma' = 1 + f_0 z^{-1} + f_1 z^{-2} + \ldots + f_v z^{-v-1}$.

The aim of the FELS is to learn the desired control action by linearly updating the control action using the output feedback error. If FELS is used alone, it can be casted as a linear controller with the following discrete transfer function

$$\frac{U(z^{-1})}{E(z^{-1})} = \frac{\gamma'}{1 - z^{-1}} \tag{3.7}$$

assuming that $t_d = 1$. Equation (3.7) includes an integrator. This implies that in the absence of integrators in the plant, the control system is only able to track constant references without incurring steady state errors. Suppose a reference model is used to generate the reference trajectory $r(t)$ from the setpoint $l(t - t_d)$ in the following manner :

$$A_m r(t) = B_m l(t - t_d) \tag{3.8}$$

Then, the constraint on the setpoint l is that it must remain constant within a period of time for the FELS to work. This implies that only steady state tracking is possible for the control system when there are no additional integrators inherent in the process.

Next, the proof of stability for the closed loop system using FELS can be shown using linear discrete analysis. The discrete transfer function of the closed loop system is

$$G_{CL}(z^{-1}) = \frac{Y(z^{-1})}{R(z^{-1})} = \frac{\gamma' B_p z^{-t_d}}{(1-z^{-1})A_p + \gamma' B_p} \tag{3.9}$$

When $G_{CL}(z^{-1})$ is stable, and given that $\lim_{t\to\infty} r(t-t_d) = l$, the application of the final value theorem on Equation (3.9) reveals that

$$\begin{aligned} \lim_{t\to\infty} y(t) &= \lim_{z\to 1}(1-z^{-1})G_{CL}(z^{-1})R(z^{-1}) \\ &= \lim_{z\to 1}(1-z^{-1})\left(\frac{\gamma' B_p z^{-t_d}}{(1-z^{-1})A_p + \gamma' B_p}\right)\left(\frac{l}{1-z^{-1}}\right) \\ &= l \end{aligned} \tag{3.10}$$

This implies that $\lim_{t\to\infty} r(t-t_d) - y(t) = 0$. Subtracting $U^*(t-t_d)$ from both sides of Equation (3.4) results in

$$B_p U^*(t-t_d) - A_p y(t) = B_p \tilde{U}(t-t_d) \tag{3.11}$$

where $\tilde{U}(t-t_d) = U^*(t-t_d) - U(t-t_d)$. Substituting Equation (3.4) into Equation (3.11),

$$A_p e(t) = B_p \tilde{U}(t-t_d) \tag{3.12}$$

Equation (3.12) shows the relationship between the output error and the estimation error in the desired control action. If A_p is stable, then the convergence of $e(t)$ will imply $\tilde{U}(t-t_d) \to 0$. Hence, the convergence of $U \to U^*$ at steady state is proved. In summary, the ability of FELS to estimate the desired control action is based on the following conditions:

1. The setpoint l remains constant for a period of time,

2. $G_{CL}(z^{-1})$ is stable for the Final Value Theorem to be applicable, and

3. A_p is stable.

With an insight into the convergence requirements for FELS, an analysis on the other update law used in the self-learning control scheme is presented next.

3.3 Stability criterion for the NLMS

In the self-learning control scheme, the role of the NLMS learning rule is to update the weights of the neurofuzzy controller using the desired control action estimated by the FELS. In this section, an analysis of the convergence properties of the NLMS algorithm assuming the availability of the required control action. The NLMS rule updates the weights $w(t)$ of the neurofuzzy controller in the following manner :

$$w(t) = w(t-1) + \frac{\delta a(t)}{a^T(t)a(t)}\tilde{U}(t-t_d) \tag{3.13}$$

where $a(t)$ is the transformed input vector, δ is the update rate, and $\tilde{U}(t-t_d) = U^*(t-t_d) - U(t-t_d)$ is the control action error at time $t-t_d$. Assuming that there exists an ideal weight vector w^*, the output control action error can be defined as

$$\tilde{U}(t-t_d) = a^T(t)(w^* - w(t-t_d)) \tag{3.14}$$

Two derivations of the stability of this update law are presented. The first method makes use of the Lynpunov's method to show that this simple update law is able to ensure that $w(t) \rightarrow w^*$. Let the Lynpunov function candidate be

$$V(t) = \tilde{w}^T(t)\tilde{w}(t) \tag{3.15}$$

where $\tilde{w}(t) = w^* - w(t)$. The rate of change of the quadratic Lynpunov function candidate can be written as

$$\begin{aligned}
\Delta V(t) &= V(t) - V(t-1) \\
&= \tilde{w}^T(t)\tilde{w}(t) - \tilde{w}^T(t-1)\tilde{w}(t-1) \\
&= (\tilde{w}(t) - \tilde{w}(t-1))^T (\tilde{w}(t) + \tilde{w}(t-1)) \\
&= (\tilde{w}(t) - \tilde{w}(t-1))^T (\tilde{w}(t) - \tilde{w}(t-1) + 2\tilde{w}(t-1)) \\
&= \Delta\tilde{w}(t)^T (\Delta\tilde{w}(t) + 2\tilde{w}(t-1))
\end{aligned} \tag{3.16}$$

where $\Delta\tilde{w}(t) = \tilde{w}(t) - \tilde{w}(t-1)$. Expanding $\Delta\tilde{w}(t)$ and making use of the NLMS

update law :

$$\begin{aligned}\Delta\tilde{w}(t) &= \tilde{w}(t) - \tilde{w}(t-1) \\ &= w^* - w(t) - w^* + w(t-1) \\ &= -\frac{\delta a(t)}{a^T(t)a(t)}\left\{\tilde{U}(t-t_d)\right\} \\ &= -\frac{\delta a(t)}{a^T(t)a(t)}\left\{a^T(t)(w^* - w(t-t_d))\right\} \end{aligned} \tag{3.17}$$

Examining Equation (3.16) term by term using Equation (3.17) produces

$$\Delta\tilde{w}^T(t)\Delta\tilde{w}(t) = \frac{\delta^2\tilde{U}(t-t_d)^2}{a^T(t)a(t)} \tag{3.18}$$

and

$$2\Delta\tilde{w}^T(t)\tilde{w}(t-1) = \frac{2\delta\tilde{U}(t-t_d)a^T(t)\tilde{w}(t-1)}{a^T(t)a(t)} \tag{3.19}$$

Combining Equations (3.18) and (3.19), Equation (3.16) becomes

$$\Delta V(t) - V(t-1) = \frac{\delta\tilde{U}(t-t_d)^2}{a^T(t)a(t)} - \frac{2\delta\tilde{U}(t-t_d)a^T(t)\tilde{w}(t-1)}{a^T(t)a(t)} \tag{3.20}$$

When $t_d = 1$, Equation (3.20) is reduced to the following:

$$\begin{aligned} V(t) - V(t-1) &= -\frac{\delta(2-\delta)\tilde{U}(t-1)^2}{a^T(t)a(t)} \\ &= -\frac{\delta(2-\delta)\left(a^T(t)\tilde{w}(t-1)\right)^2}{a^T(t)a(t)} \end{aligned} \tag{3.21}$$

Equation (3.21) is negative if $0 \leq \delta \leq 2$, which is consistent with stability condition for the NLMS update algorithm. However, the derivation proof is restrictive in the sense that it is only valid for a delay of 1 sample. Next, an alternative proof is presented based on linear discrete analysis in a bid to overcome this restriction.

First, multiplying $a^T(t)$ to both sides of Equation (3.13) results in

$$\begin{aligned} a^T(t)\left(w(t) - w(t-1)\right) &= \delta\tilde{U}(t-t_d) \\ &= \delta a^T(t)\left(w^* - w(t-t_d)\right) \end{aligned} \tag{3.22}$$

When Equation (3.22) is re-arranged, a relationship between $w(t)$ and w^* is found as

$$\left(1 - z^{-1} + \delta z^{-t_d}\right) w(t) = \delta w^* \tag{3.23}$$

From linear discrete analysis, the condition for stable weight updates is

$$|z| < 1 \tag{3.24}$$

From this condition, the constraint on δ can be determined by finding the roots of $(1 - z^{-1} + \delta z^{-t_d})$. For the simple case when $t_d = 1$, the constraint shown in Equation (3.24) becomes

$$0 \leq \delta \leq 2 \tag{3.25}$$

which is the same as the derivation based on the Lynpunov's theorem earlier. Next, the convergence of $w(t) \rightarrow w^*$ can be shown using Equation (3.23) and the final value theorem :

$$\begin{aligned} \lim_{t\to\infty} w(t) &= \lim_{z\to 1}(1 - z^{-1}) \left(\frac{\delta}{1 - z^{-1} + \delta z^{-t_d}}\right) \left(\frac{w^*}{1 - z^{-1}}\right) \\ &= w^* \end{aligned} \tag{3.26}$$

assuming that w^* is constant.

If $w(t)$ eventually converges to w^* ($1 - z^{-1} + \delta z^{-t_d}$ is stable), substituting Equations (3.14) into (3.12) reveals

$$A_p e(t) = B_p a^T(t) \left(w^* - w(t - t_d)\right) \tag{3.27}$$

Equation (3.27) implies that if A_p is stable, then $e(t) \rightarrow 0$ and the control objective is achieved.

In conclusion, a scheme using the NLMS rule to update the weights of the neurofuzzy feedforward controller should satisfy the following conditions :

1. $1 - z^{-1} + \delta z^{-t_d}$ must be stable

2. A_p is stable

The first condition restricts the update rate δ of the weights, while the second condition places restrictions on the type of process that the control scheme can control. In classical control, the restriction on A_p stable implies that the plant is stable or must be stabilized before applying control.

However, there is an important factor that determines the rate with which $w(t)$ converges to w^*. Specifically, the condition number C is a gauge of the rate of convergence of the algorithm:

$$C = \frac{\max(eig(R))}{\min(eig(R))} \tag{3.28}$$

where R is the autocorrelation matrix of $a(t)$. When the condition number is 1 and $\delta = 1$, instantaneous convergence of $w(t)$ is obtained. On the other hand, with large values of C, slow convergence will occur, which in turn may lead to numerical difficulties that may even cause the adaptive algorithm to be unstable. This can be illustrated by the most extreme case of an infinite C, which means that the transformed vector $a(t)$ is always zero for some weights. This fact in turn means that the weights will never get updated, and thus will never converge to their desired value. This condition is commonly known as *persistent excitation*.

Having gained an insight to the two optimization strategies used in the self-learning control scheme, the next section analyzes the stability criterion for the self-learning control scheme as a whole.

3.4 Stability Criterion for the Self-learning Control Scheme

The derivation of the stability criterion for the self-learning control scheme is also based on linear discrete analysis since it is shown in the previous section that there is difficulty in applying the conventional Lynpunov's stability technique. The output of the self-learning controller is a combination of the output from the neurofuzzy controller and the proportional controller :

$$\begin{aligned} U(t-t_d) &= u_f(t-t_d) + u_b(t-t_d) \\ &= a^T(t)w(t-t_d) + k_p e(e-t_d) \end{aligned} \tag{3.29}$$

The aim of the on-line training mechanism is to enable the neurofuzzy controller to learn the inverse plant dynamics. This control law when combined with the plant

dynamics defined in Equation (3.1) becomes

$$\left(A_p - B_p k_p z^{-t_d}\right) e(t) = B_p a^T(t) \left(w^* - w(t - t_d)\right) \tag{3.30}$$

Combining the two optimization strategies, namely the FELS (Equation (3.6)) and the NLMS algorithm (Equation (3.13)), the following update law is obtained :

$$\begin{aligned} w(t) &= w(t-1) + \frac{\delta a(t)}{a^T(t)a(t)} \left(u_f(t - t_d) + \gamma' e(t) - a^T(t) w(t-1)\right) \\ &= w(t-1) + \frac{\delta a(t)}{a^T(t)a(t)} \left(\gamma' e(t) - a(t) \left(w(t-1) - w(t - t_d)\right)\right) \end{aligned} \tag{3.31}$$

Multiplying $a(t)$ to the left hand side of Equation (3.31) results in

$$\alpha a^T(t) w(t) = \delta \gamma' e(t) \tag{3.32}$$

where $\alpha = 1 - (1 - \delta) z^{-1} - \delta z^{-t_d}$. Now, substituting Equation (3.30) into Equation (3.32) :

$$(A_p - B_p k_p z^{-t_d}) \alpha a^T(t) w(t) = B_p \delta \gamma' a^T(t) \left(w^* - w(t - t_d)\right) \tag{3.33}$$

or equivalently

$$\left((A_p - B_p k_p z^{-t_d}) \alpha + B_p \delta \gamma' z^{-t_d}\right) w(t) = B_p \delta \gamma' w^* \tag{3.34}$$

From Equation (3.34), the optimization strategies will be stable provided that the magnitude of the roots of $((A_p - B_p k_p z^{-t_d})\alpha + B_p \delta \gamma' q - t_d)$ is less 1. This stability criterion is derived without using any additional restriction or assumption. Next, the convergence of the weights towards their ideal values is demonstrated using the final value theorem. Applying the theorem to Equation (3.34) results in

$$\begin{aligned} \lim_{t \to \infty} w(t) &= \lim_{z \to 1} (1 - z^{-1}) \left(\frac{B_p \delta \gamma'}{(A_p - B_p k_p z^{-t_d})\alpha + B_p \delta \gamma' z^{-t_d}}\right) \left(\frac{w^*}{1 - z^{-1}}\right) \\ &= w^* \end{aligned} \tag{3.35}$$

which verifies that the $w(t) \to w^*$ when w^* is constant. Similar to the idealized case for NLMS, the convergence of the weights is dependent upon the condition number of $a(t)$ as in Equation (3.28). In the self-learning control scheme, $a(t)$ is fixed when the inputs to the neurofuzzy model are the reference trajectory and

its history (see Equation (2.3)). This condition places constraints on the choice of the reference model and the B-spline network parameters like the position of the knots and the number of sets for each input, etc. Improper choices of these parameters can lead to very large condition numbers that may result in extremely slow convergence.

If the requirements for $w(t)$ to converge is satisfied, Equation (3.30) indicates that the output error $e(t)$ will eventually decay to zero from if $A_p - B_p k_p z^{-t_d}$ is stable. Thus, the control objective will be met. Also, the convergence of FELS is obvious if $w(t) \rightarrow W^*$ from Equation (3.29).

To summarize, the requirements for the convergence of the self-learning control scheme are listed below:

$$|(A_p - B_p k_p q{-}t_d)\alpha + B_p \delta \gamma' z^{-t_d}| < 1 \tag{3.36a}$$

$$|A_p - B_p k_p z^{-t_d}| < 1 \tag{3.36b}$$

Next, the feasibility of these conditions is tested via simulations.

3.4.1 Simulation Verification

In this section, simulation results are presented to verify the proposed stability requirements. The study makes use of the following first order plus dead-time plant (Tan and Lo, 2001*a*) :

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K_g e^{-\tau_d s}}{\tau s + 1} \tag{3.37}$$

where $K_g = 20$, $\tau = 150$ and $\tau_d = 35$. Using a sampling period of 5 seconds, the delay (with Zero-Order Hold incorporated) of the discretized system is 8 sampling instants. Listed below are the parameters of the discrete plant :

$$A_p = 1 - 0.9672 z^{-1} \tag{3.38a}$$

$$B_p = 20 \tag{3.38b}$$

As the plant is a first order model, the inputs of the neurofuzzy controller are chosen as $r(t)$ and $\Delta r(t)$ respectively. They are each spanned by two triangular

fuzzy sets with apexes at [5,15] and [-0.5,0.5] respectively. The reference model used is a first order model with a time constant of 100 seconds, and the setpoint is set to alternate between 5 and 15 every 20 minutes. These choices lead to a relatively small condition number of 80 for the transformed input vector $a(t)$. Ziegler Nichols tuning rules suggest a gain (K) and integral time (T_i) of 0.193 and 116.67 seconds for a PI controller respectively. The parameters of the control scheme are commissioned using Equation (2.43) with the update rate of the NLMS algorithm as unity ($\delta = 1$) :

$$\begin{aligned} \gamma &= \frac{1}{K_g} = 0.05 \\ \lambda_1 &= \frac{Kt_d}{\gamma} = 30.1959 \\ k_p &= \frac{Kht_d}{T_i} - \gamma = 0.161 \end{aligned} \tag{3.39}$$

The investigations into the effectiveness of the conditions are carried out by increasing each learning parameter in turn until the limit of the stability conditions is reached. Table 3.1 summarizes the results of the study. From the table, it can be seen that the stability criteria works reasonably well. The prediction scheme is slightly conservative for the case when $k_p = 0.3$.

Despite the establishment of the stability and convergence conditions for the self-learning control scheme, there is difficulty implementing these checks in practice as the stability criterion in Equation (3.34) requires full plant knowledge. If extensive modelling is needed to obtain these plant parameters, then it defeats the purpose of having a self-learning control scheme when all of the plant's parameters are known. Although there are no clear extensions of these stability conditions when nonlinear plants are used, linearized nonlinear plant dynamics may be used together with the stability criteria to provide some idea about system stability.

3.5 Conclusion

This chapter provides an insight into the stability of a neurofuzzy self-learning control scheme using linear discrete analysis. Stability criterions are imposed onto

Change	Max |root| of $(A_p - B_p k_p z^{-t_d})\alpha + B_p \delta \gamma' z^{-t_d}$	Max |root| $A_p - B_p k_p z^{-t_d}$	Stable
nominal	.91796	.97942	Yes
$k_p = 0.3$	1.0508	1.0811	Yes
$k_p = 0.4$	1.0655	1.1009	No
$\gamma = 0.09$	.95580	.97942	Yes
$\gamma = 0.1$	1.0037	.97942	No
$\lambda_1 = 59$	.97675	.97942	Yes
$\lambda_1 = 60$	1.0009	.979426	No

Table 3.1. Summary of the simulations performed

the learning parameters of the control scheme, and simulation results are presented to verify the effectiveness of the criteria.

Chapter 4

Neurofuzzy Control of a Liquid Level Process

4.1 Introduction

The proposed feedback error learning strategy has been demonstrated to have superior learning rates when compared to the original and modified learning strategies (Tan, 1997; Santos *et al.*, 2000) for first and second order linear plants (Lo, 2001). In this chapter, the neurofuzzy control scheme using the various feedback error learning rules are evaluated on a simulated liquid level plant that has been used as a test bed for many non-linear control strategies (Postlethwaite *et al.*, 1997; Edgar and Postlethwaite, 2000; Linkens and Kandiah, 1996). The idea is to extend the analysis of the feedback error learning rules to a nonlinear plant. Furthermore, experiments are carried out on a actual liquid level plant to test the feasibility of using the proposed feedback error learning strategy (FELS) with the neurofuzzy control scheme.

The organization of this chapter is as follows. First, the liquid level plant is described. This is followed by a description of the control scheme design. Simulation results are then presented to compare between various learning rules. Lastly, experiments are carried out to prove the feasibility of the proposed FELS, as described in Section 4.5.

4.2 The Liquid Level Process

The task is to control the liquid level in a uniform cross-section tank as shown in Figure 4.1. Control of the liquid level is achieved by pumping water into the tank from the top, while water leaves the tank via a hole at the bottom of the tank.

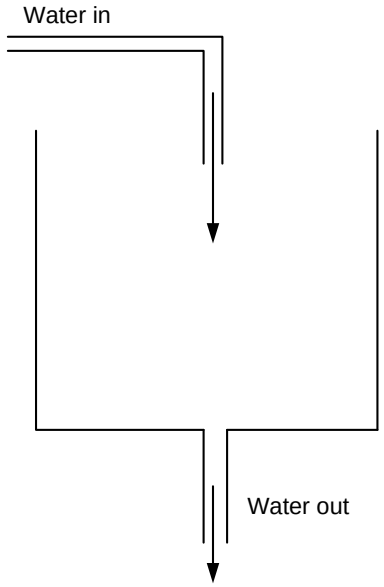

Figure 4.1. The simulated liquid level plant

The mathematical model of this process is a single, non-linear differential equation of the form:

$$\rho A \frac{dh_L}{dt} + \beta\sqrt{h_L} = F_i \tag{4.1}$$

where $\rho = 1gcm^{-3}$ is the density of water,

$A = 10cm^2$ is the horizontal cross-sectional area,

$\beta = 1gs^{-1}cm$ is a flow coefficient,

h_L is the liquid level measured in cm, and

F_L is the inlet liquid flow rate measured in gs^{-1}.

The nonlinearity arises from the square root of the height, h_L, in the differential equation. In physical terms, the nonlinearity arises from the relationship between velocity and pressure drop in the liquid level, and is commonly known as the

simplified Bernoulli equation for incompressible fluid (Seborg *et al.*, 1989). The linearized dynamics of the system is

$$\tau \frac{dh_L^*}{dt} + h_L^* = K_f F_i^* \tag{4.2}$$

where $\tau = \frac{2\rho A\sqrt{h_{ss}}}{\beta}$ is the time constant

$K_f = \frac{2\sqrt{h_{ss}}}{\beta}$ is the static gain of the linearized model

$*$ denotes the deviation variable

h_{ss} is the liquid level at the point of linearization.

The nonlinearity in the plant's dynamics is clearly evident in Figure 4.2, which shows the plots of the linearized gains and time constants at various liquid levels. Suppose the liquid level is only allowed to vary between 0 and 100cm, and the control problem that is tackled is to provide good servo control when the set point is varied near the bottom (between 10 and 15 cm), as well as near the top of the tank (between 90 and 95 cm). Between the two operating regions, the system characteristics vary approximately by a factor of three, as seen in Figure 4.2. The rate change in the system parameters is also higher when the liquid level is low. Consequently, the control problem at the bottom of the tank may be tougher in comparison to control at the top.

With this knowledge about the plant, the next step is to formulate the self-learning control scheme to achieve good control performance.

4.3 Neurofuzzy Controller Design

This section presents the design of the neurofuzzy model and the parameters used in the evaluation of the feedback error learning strategies. The linearized system dynamics in Equation (4.2) shows that the liquid level process may be modelled as a first order system. Therefore, the input vector that is needed by the neurofuzzy controller to model the inverse plant dynamics (from Equation (2.3)) is

$$\mathbf{x}(t) = [r(t+1) \quad \Delta r(t+1)]^T \tag{4.3}$$

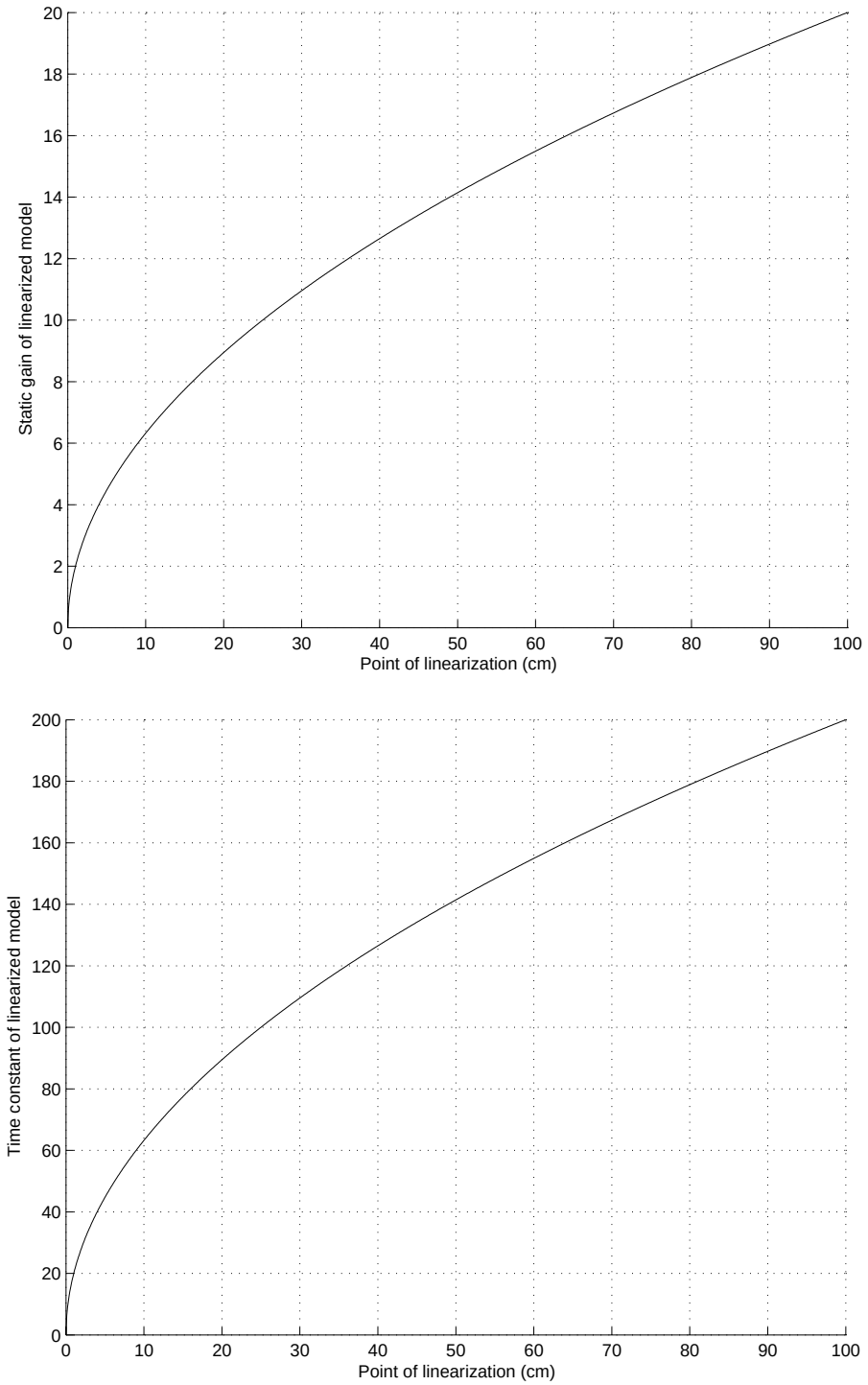

Figure 4.2. Linearized gain and time constant of the liquid level plant

The need for the feedforward controller to predict one sampling interval ahead in time is to compensate for the delay resulting from the use of a zero-order hold to convert the sequence of digital control action into a continuous time signal.

Since the liquid level is constrained between 0 and 100cm, five fuzzy sets, centered at 0, 25, 50, 75 and 100cm respectively, are used to characterize the input space of the reference signal $r(t+1)$. As the change between the set points is 5cm, the lower and upper bound of the universe of discourse for the rate of change of the reference signal ,$\Delta r(t+1)$, is set at -5 and 5 respectively. The input space for $\Delta r(t+1)$ is then partitioned by three fuzzy sets with apexes at -5, 0 and 5. All the weights in the neurofuzzy model are initialized to zero to simulate the situation in which no prior knowledge is used.

Next on the list in the design of the control scheme is the reference model. Assuming that the constraints on the minimum and maximum flow rates are 0 and $15gs^{-1}$ respectively, and the sampling period used is 10 seconds (Postlethwaite *et al.*, 1997; Edgar and Postlethwaite, 2000; Linkens and Kandiah, 1996), the liquid level process can be shown to be only able to follow a first order reference model that has a time constant of more than 7.7 seconds to prevent the minimum flow rate constraint of 0 from being violated (Tan, 1997). Thus, the time constant of the reference model chosen in this evaluation of the learning strategies is 8 seconds.

The commissioning strategies for the various feedback error learning strategies described in Chapter 2 are used to provide the tuning parameter values, which are described in the following sections.

4.3.1 Parameters using the original FELS

Postlethwaite (1993) showed that a gain and integral time of 0.93 and 76 seconds respectively minimize the Integrated Absolute Error (IAE) when the process is operating around a level of 15 cm while providing stable control near the bottom and near the top of the tank. Setting the update rate of the NLMS algorithm to be unity, i.e, $\delta = 1$, and substituting the PI values into the commissioning strategy for the original feedback error learning rule (Equation (2.24)), the parameters of

the self-learning controller is found to be :

$$\begin{aligned} k_p &= K\left(1-\frac{h}{2T_i}\right) = 0.8688 \\ \gamma &= \frac{Kh\left[1+\delta(t_d-1)\right]}{\delta T_i} = 0.1224 \end{aligned} \tag{4.4}$$

4.3.2 Parameters using the modified FELS

When the liquid level is at 15 cm, the linearized plant dynamics in Equation (4.2) suggests that the gain k_f and time constant τ of the plant are 7.746 and 77.4597 seconds. The approximate relationship between a PI controller and the modified learning strategy proposed by Santos *et al.* (2000) (Equation (2.28)) gives

$$\begin{aligned} \lambda_1 &= \tau = 77.4597 \\ k_p &= K\left(1-\frac{h+2\lambda_1}{2T_i}\right) = -0.079 \approx 0 \\ \gamma &= \frac{Kh\left[1+\delta(t_d-1)\right]}{\delta T_i} = 0.1224 \end{aligned} \tag{4.5}$$

The proportional gain k_p as suggested by the modified learning strategy's commissioning strategy is negative, which is contradictory to the known fact that the gain used cannot be opposite to that of the process gain. Otherwise, the resulting system may be unstable. The negative k_p value is due to the second term in Equation (2.28), $\frac{h+2\lambda_1}{2T_i}$, being larger than unity (Lo, 2001). Since the gain is negative, k_p is set to zero, which effectively removes the feedback controller from the control scheme. One potential problem that will be faced when the control scheme does not have a feedback controller is that the initial control performance will be poor, as the feedforward controller is untrained. As stability may also be an issue, the stability criteria derived in Chapter 3 is used to predict if the closed loop system will remain stable. Substituting the linearized plant dynamics at the liquid height of 15 cm into Equation (3.36) :

$$\begin{aligned} \text{max root of } (A_p - B_p k_p q{-}t_d)\alpha + B_p\delta\gamma' z^{-t_d} &= 8.1180 > 1 \\ \text{max root of } A_p - B_p k_p z^{-t_d} &= 0.8789 < 1 \end{aligned} \tag{4.6}$$

where $A_p = 1 - 0.8789z^{-1}$, $B_p = 0.9381$, and $\alpha = 1 - z^{-1}$. Equation (4.6) shows that the parameters suggested by the commissioning strategy will lead to instability, and simulations performed on the liquid level plant confirmed this fact (see Figure 4.3). The failure of the commissioning strategy prevents further evaluation on the performance of the modified FELS, and thus will not be discussed further.

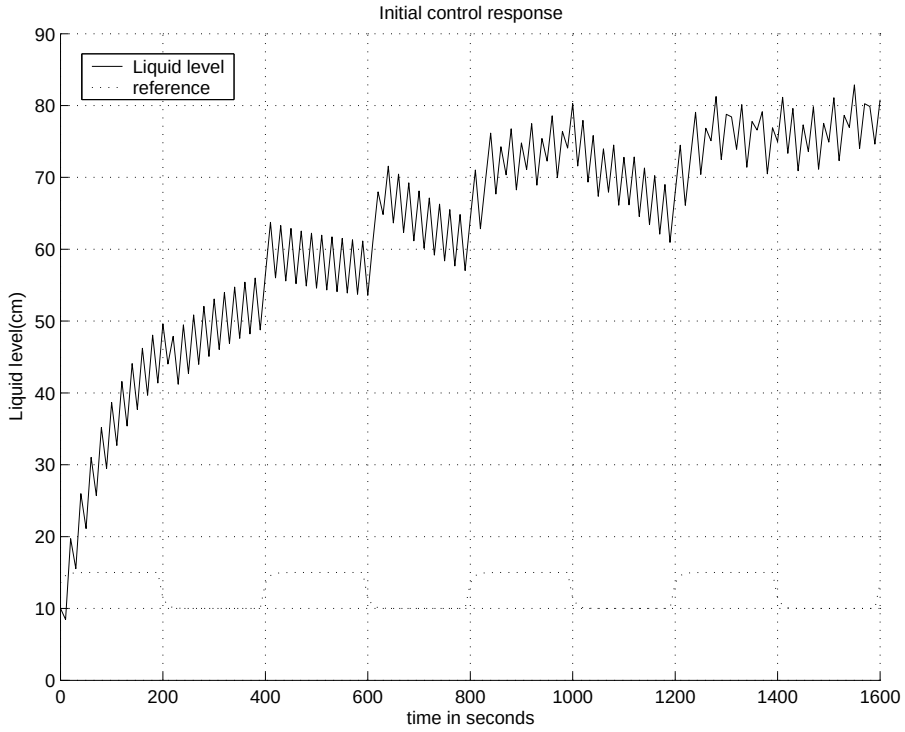

Figure 4.3. Control performance of the modified FELS

4.3.3 Parameters using the proposed FELS

For the proposed feedback error learning strategy, the commissioning strategy (Equation (2.43)) suggests the controller's parameters should be chosen as

$$\begin{aligned} \gamma &= \frac{1}{K_f} = 0.1291 \\ \lambda_1 &= \frac{Kt_d}{\gamma} = 7.2037 \\ k_p &= \frac{Kht_d}{T_i} - \gamma = -0.0067 \approx 0 \end{aligned} \tag{4.7}$$

Though the proposed feedback error learning strategy alleviates the restriction on the modified learning rule, it is obviously not sufficient as the suggested gain of the feedback controller is still negative. Like the modified learning rule, k_p is set to zero in the analysis. Evaluating the stability criteria (Equation (3.36)) with the same linearized plant dynamics as in Equation (4.6) shows

$$\begin{aligned} \text{max root of } (A_p - B_p k_p q{-}t_d)\alpha + B_p \delta \gamma' z^{-t_d} &= 0.8780 < 1 \\ \text{max root of } A_p - B_p k_p z^{-t_d} &= 0.8789 < 1 \end{aligned} \tag{4.8}$$

Unlike the modified FELS, the parameters suggested are stable, and thus allows for the evaluation of its performance (see Figure 4.4).

The learning parameters obtained using the commissioning equations that equated λ_1 to τ, as derived in Chapter 2, are

$$\begin{aligned} \lambda_1 &= \tau = 77.4597 \\ \gamma &= \frac{Kt_d}{\lambda_1} = 0.0120 \\ k_p &= \frac{Kht_d}{T_i} - \gamma = 0.1104 \end{aligned} \tag{4.9}$$

Unlike the case where $\lambda_1 = \frac{1}{K_g} = 7.2037$, k_p is positive. Next, the simulation results are described.

4.4 Simulation Results

Figure 4.4 shows a comparison of the initial control response performance between the proposed FELS and the original learning strategy. It can be seen that both commissioning methods of the proposed strategy have similar initial responses. Unlike the original learning strategy, the responses obtained using the proposed strategy does not suffer from sharp decreases in the liquid level at the bottom of the tank, except during the first epoch where the weights are untrained. Since the update rate, δ, for the weight vector is set to unity, the weight vector will be updated in such a way the control error for that instant is eliminated. Thus, the quality of the estimated control action plays a crucial role in how the weights

are updated, and the results demonstrate that the proposed learning strategy's estimation is far superior to that of the original strategy. The proposed learning rule results in a smaller overshoot and undershoot compared with the original strategy. Thus, the proposed learning strategy is more desirable in systems where overshoot and undershoot may be less tolerable.

Despite supposedly having a better learning capability, the proposed strategy has a poorer performance in the first epoch. This is due to the presence of unlearned weights in the neurofuzzy model, which will cause the proportional controller in the self-learning control scheme to dictate the performance of the overall system. Thus, due to the fact that the proportional gains k_p set by both proposed methods are smaller in magnitude compared with the original strategy, the slightly poorer performance of the control schemes using the proposed feedback error learning strategy is then expected.

From Figure 4.5, it can be seen in the plot of the final responses that all the three cases are able to follow the reference trajectory well. This implies that the control scheme using the proposed feedback error learning strategy is able to estimate the desired control action, which is similar to the case for the original learning strategy. However, the advantage of the proposed learning strategy over the original can be seen in the comparison of the Integral Absolute Error (IAE) computed over successive periods of 800 seconds, which is plotted in Figure 4.6. From the figure, it can be seen that even though the control schemes using the proposed learning strategy has a much poorer IAE value in the first cycle, their IAE values converge at a faster rate so as to overtake the original relationship by the second epoch. This is due to the ability of the proposed learning strategy to estimate the required control action at the faster rate than the original.

Thus, a conclusion of this study is that although the initial performance of the proposed learning strategy may be poorer due to the smaller proportional gain used in the control scheme, the faster learning rate provided by the proposed feedback error learning strategy enables the neurofuzzy controller to converge to the 'learned' response significantly faster compared with the original strategy. However, the

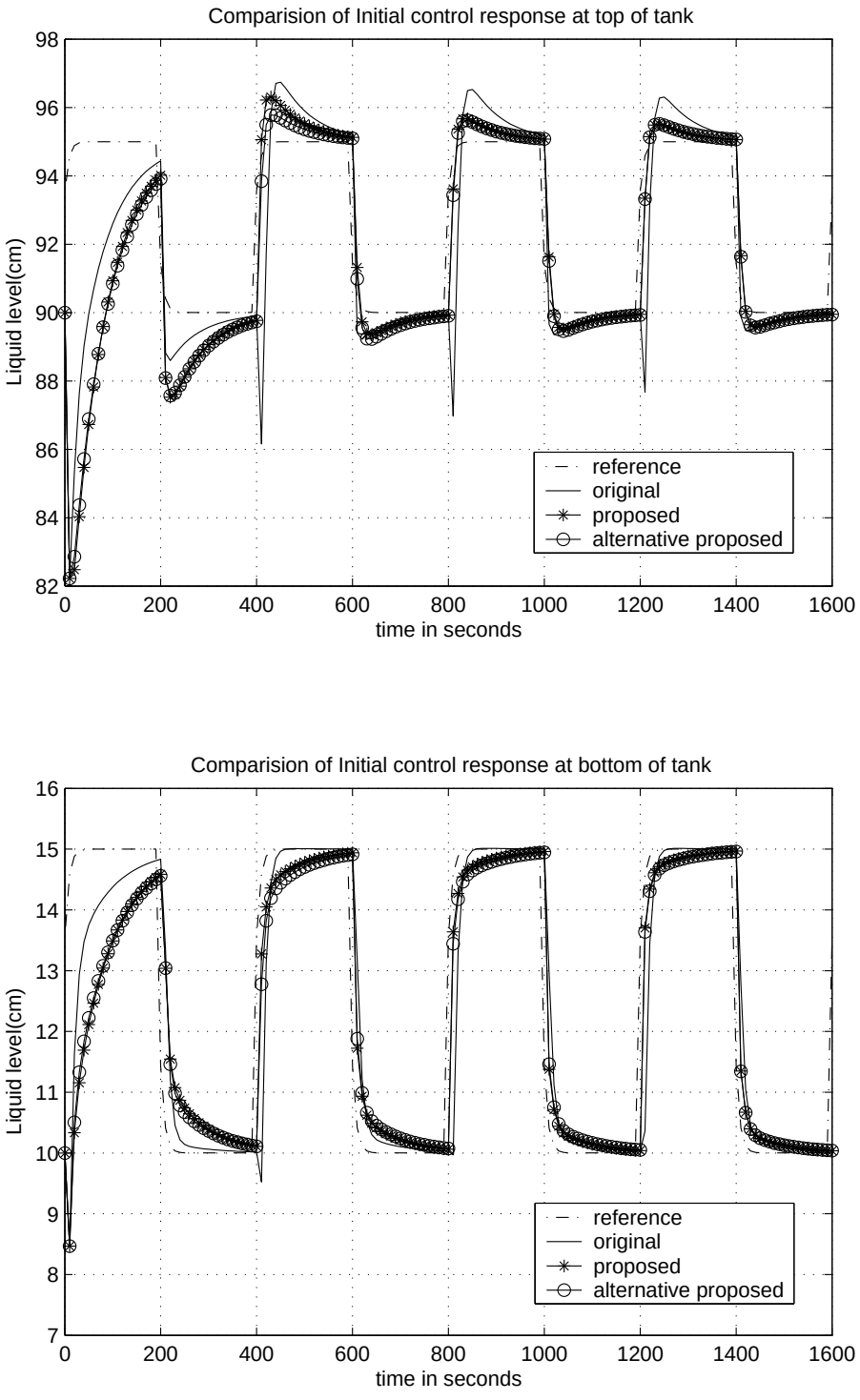

Figure 4.4. Comparison of initial response of various strategies

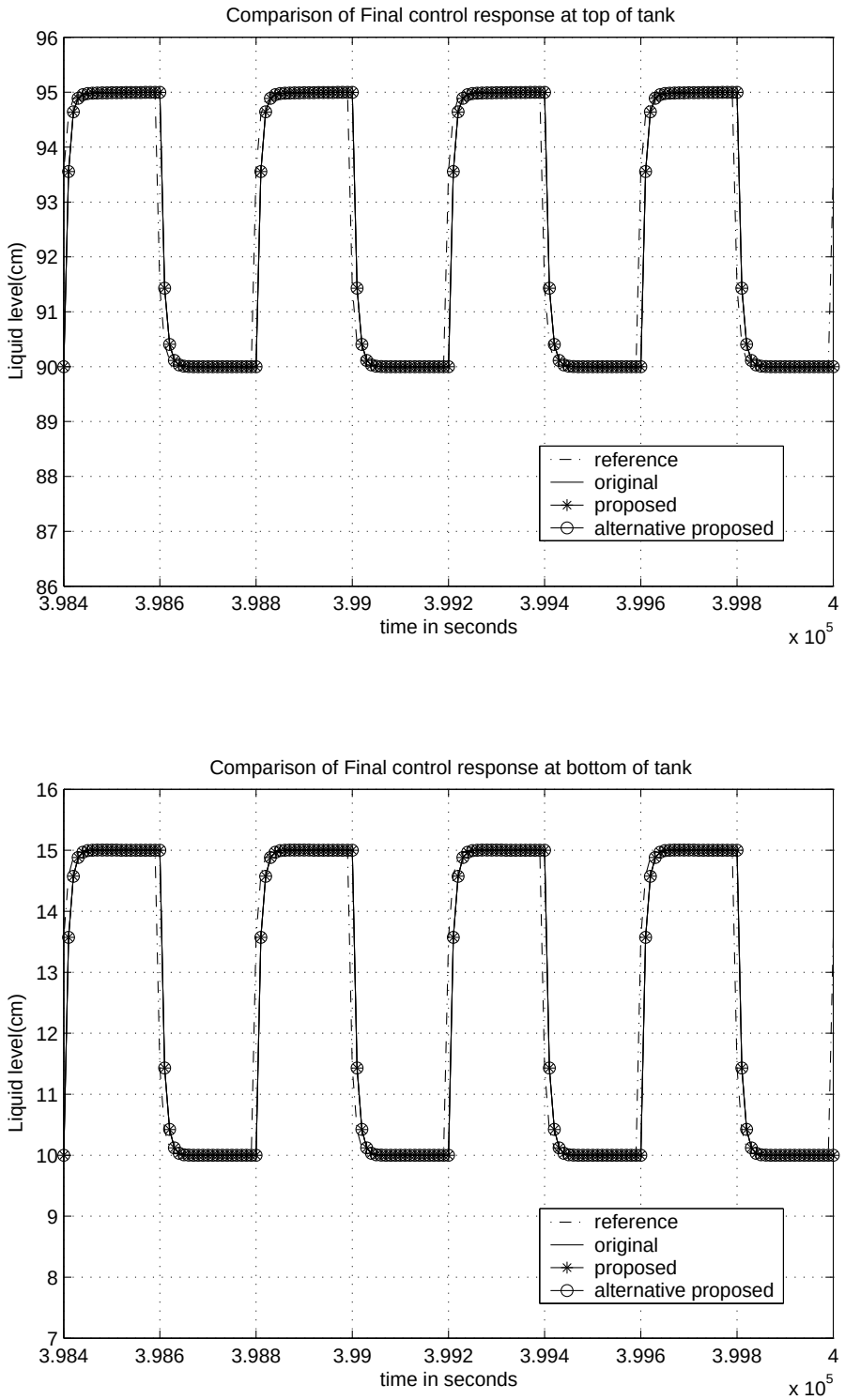

Figure 4.5. Plot of 'learned' response for the original and proposed learning strategies

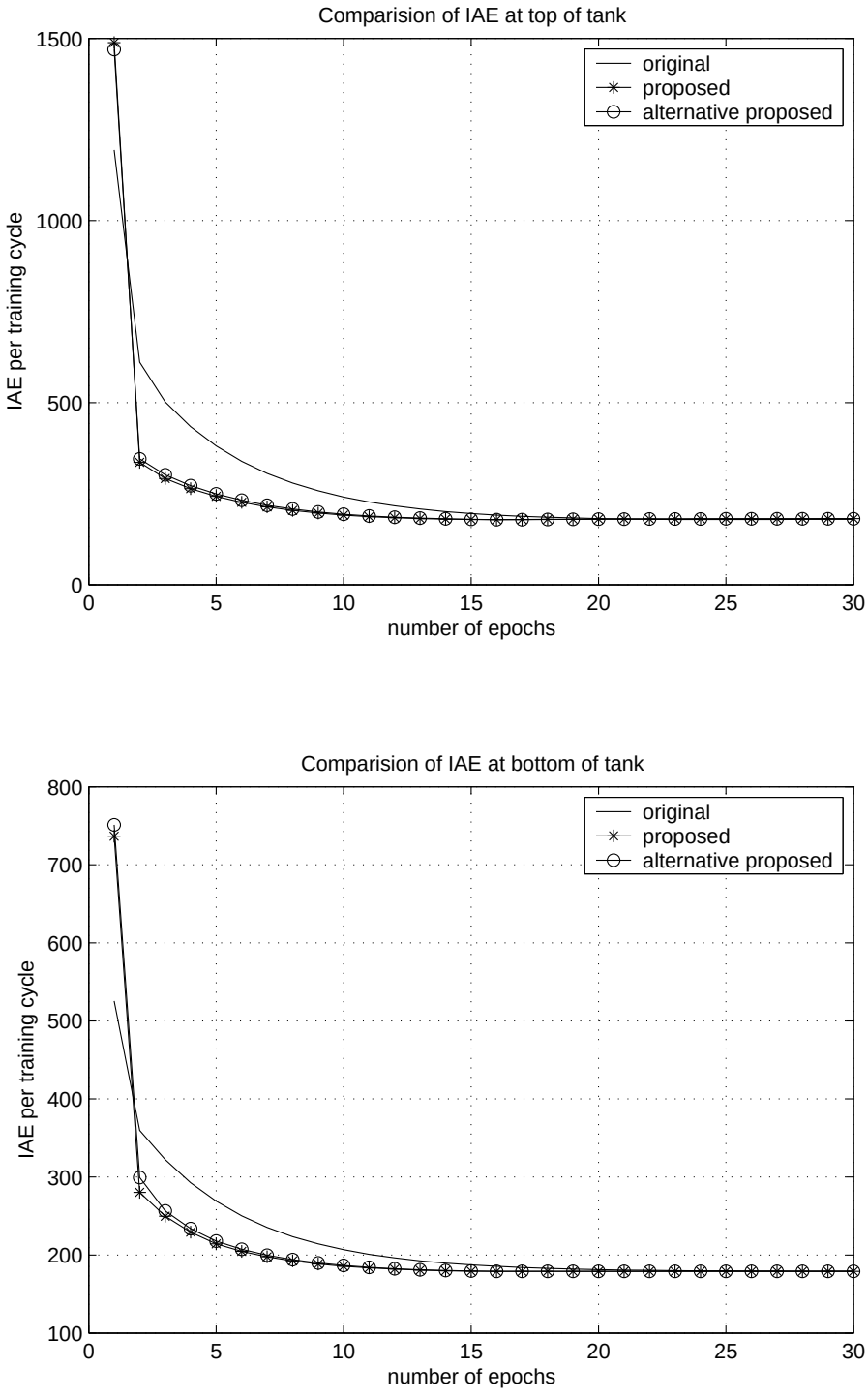

Figure 4.6. Comparison of IAE between the original and proposed learning strategies

close similarity in the performances between the two methods of commissioning the control scheme using the proposed learning strategy makes it difficult to draw any conclusion about the differences between them. This may be expected since both commissioning methods are derived using similar methodology and the same PI parameters are used for initialization. To explore the proposed learning rules further, the performance of a self-learning controller which employs a reference model that cannot be tracked by the neurofuzzy model is presented in the next section.

Effects of using a non-trackable reference model

In practice, it may be difficult to know the exact structure of the neurofuzzy model that should be used to model the inverse process dynamics. The usage of a reference model that is impossible for the system to track due to physical constraints will introduce modelling mismatches into the system. In this section, the purpose is to determine the effects of using the proposed learning strategy under this situation. As previously mentioned, the liquid level process is unable to follow a reference model with a time constant that is less than 7.7 seconds. Thus, the reference model is now arbitrarily set to 4.5 seconds, and the simulations described in the previous section are repeated.

The major difference in the performance with a reference trajectory which cannot be followed by the plant is seen when the neurofuzzy model has supposedly "learned" the inverse dynamics of the plant. As shown in Figure 4.7, the "learned" system performance deteriorated when the liquid level is low. This is due to the constraint in the control action output. As shown in Figure 4.8 for the proposed strategy, the feedforward controller's control action output cannot be realized. To verify that the rate of flow constraint is indeed the cause for the poor performance, the constraint is removed, and Figure 4.9 shows that the set point can be tracked perfectly by the proposed learning strategy.

When modelling errors are present, the proportional controller in the control scheme plays an important role in minimizing the amount of error. For the pro-

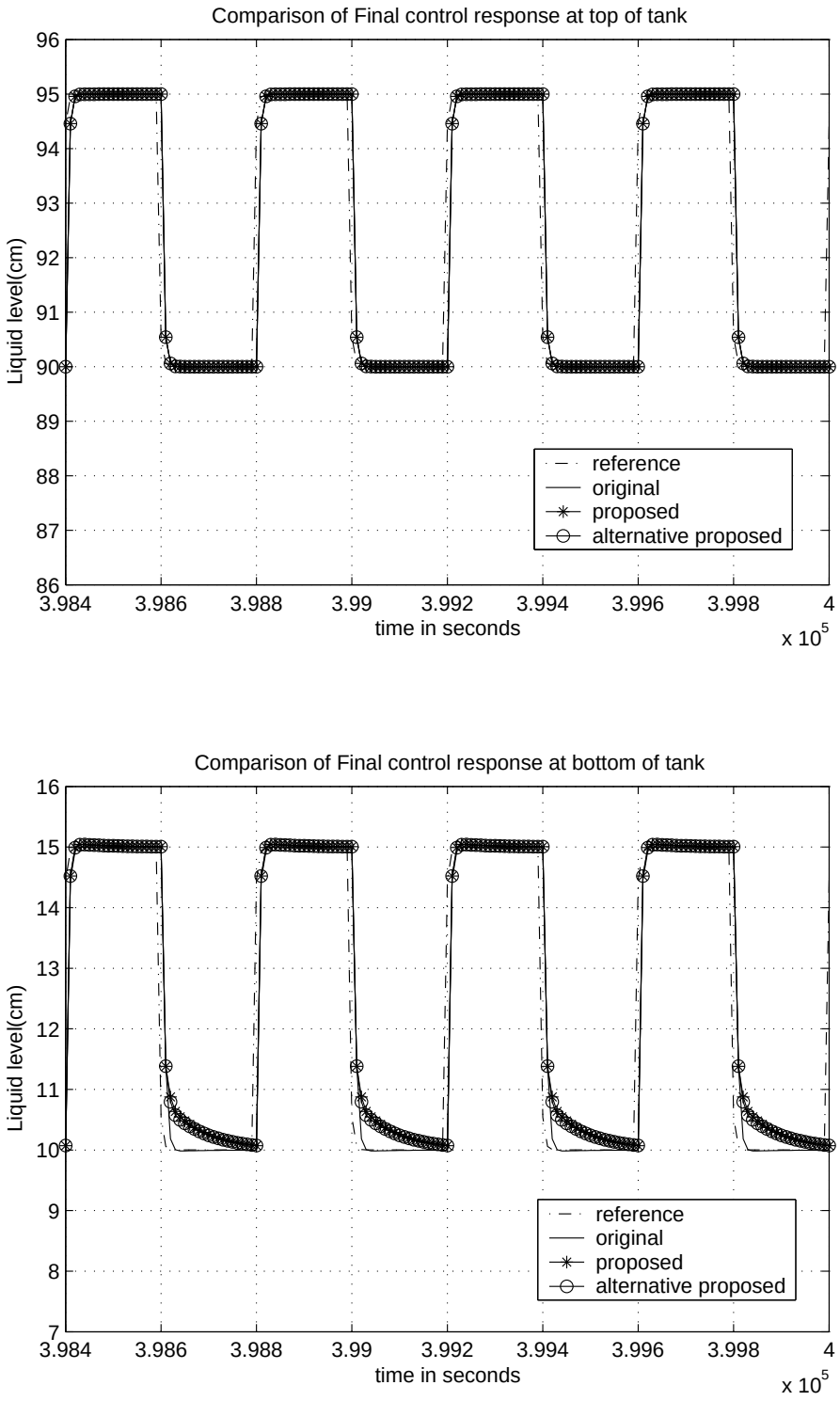

Figure 4.7. Final system response when reference trajectory is not trackable

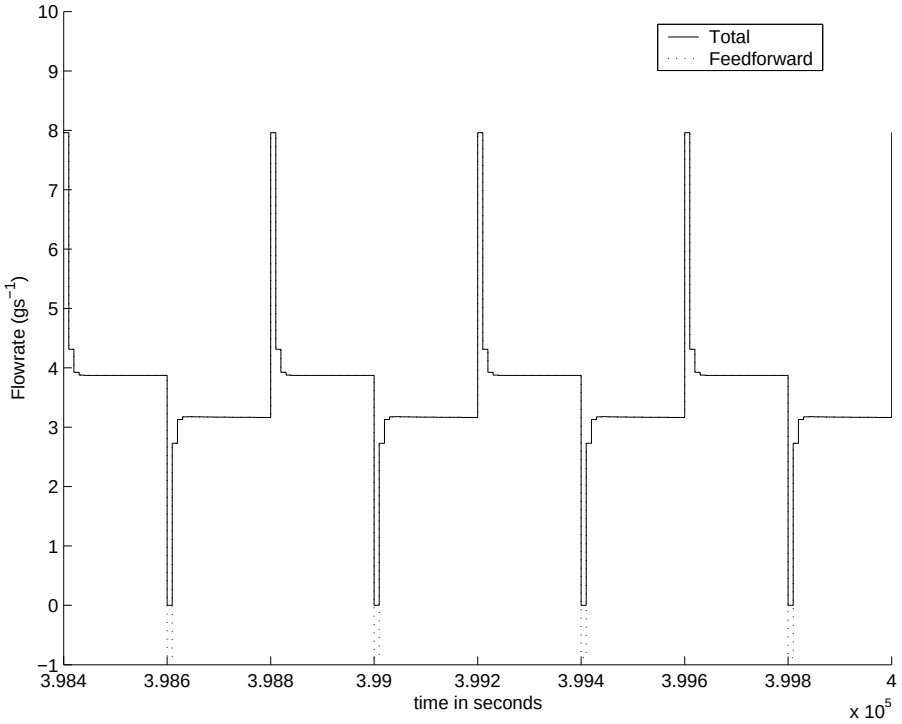

Figure 4.8. Final control action when reference trajectory is not trackable

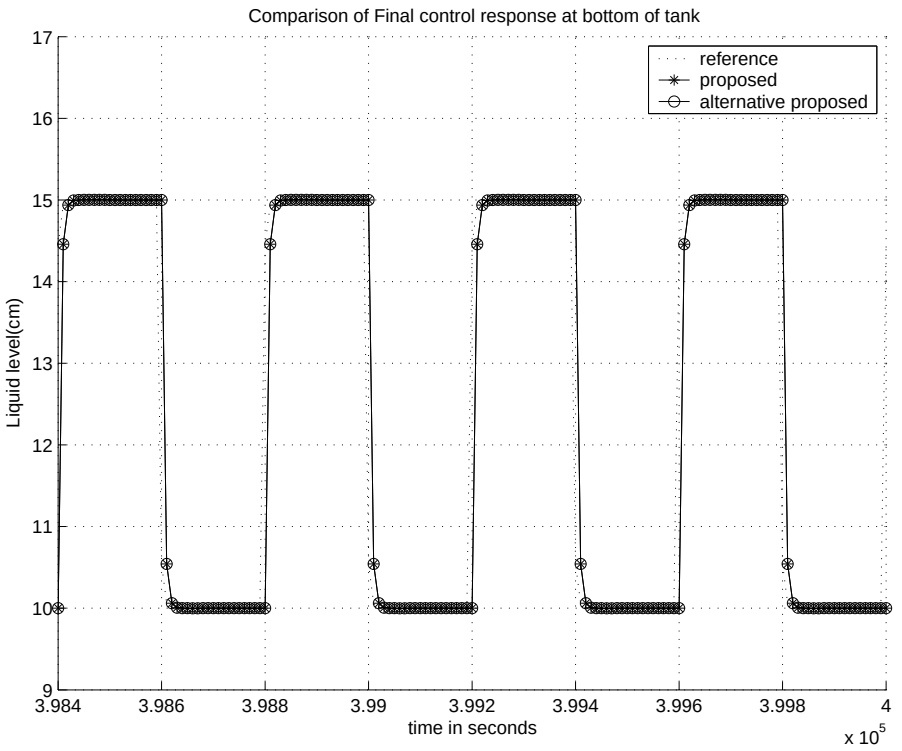

Figure 4.9. Final control response with flow rate constraint removed

posed feedback error learning strategy, the amount of feedback gain suggested is smaller in magnitude compared with the original strategy. This accounts for the poorer steady state performance using the proposed strategy compared with the original strategy. To demonstrate the influence of the proportional controller on the control performance, the proportional controller is removed from the control scheme by setting its gain to zero and re-simulated for the case using the original feedback error learning strategy at the bottom of the tank. The "learned" output response of the control scheme is shown in Figure 4.10. It is observed by comparing Figure 4.7 and Figure 4.10 that the performance has deteriorated as overshoot is present in step decreases for control at the bottom of the tank. The steady state IAE value here is similar to the proposed feedback error learning strategy's at 219. Compared with the IAE value of 144 when the proportional controller is present, this clearly shows the need for the proportional controller when there are modelling mismatches to minimize the output error is clear.

The results indicate that the control scheme using the proposed feedback error learning strategy performs poorly compared to the original feedback error learning strategy because the proportional gain suggested by the commissioning rule for the proposed strategy is far smaller than that suggested by the approximate relationship between a PI controller and the original FELS. In conclusion, the amount of proportional gain used affects the overall performance significantly if modelling errors are present, as the neurofuzzy model cannot emulate the inverse plant dynamics. In the next section, the self-learning control scheme using the proposed FELS is experimentally demonstrated.

4.5 Experimental control of a liquid level plant

This section aims at verifying the proposed FELS experimentally. First, the experimental setup is described. This is followed by a description of how the controller parameters were chosen, and the simulated results using these choices. Finally, experiments are conducted and the results are presented.

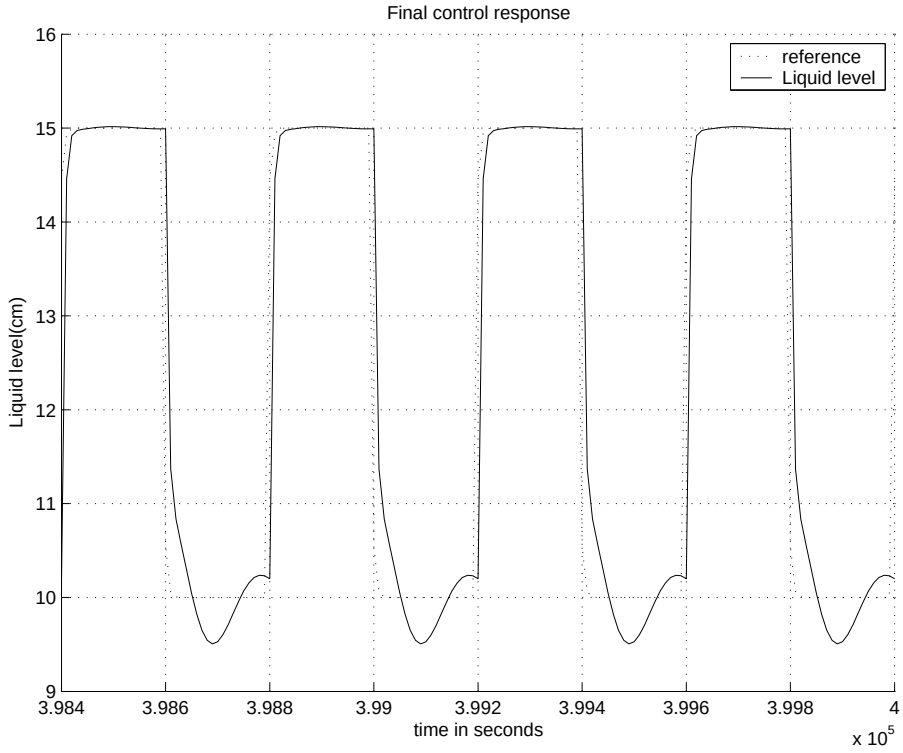

Figure 4.10. Final response of system using the original strategy and without a proportional controller

4.5.1 Experimental Setup and Plant characterization

The experimental setup consists of a Kent Ridge Instruments Coupled Tank PP-100 as the liquid level plant, a National Instruments data acquisition card (LAB-PC-1200) to acquire the data, and a computer acting as the controller. As shown in the schematic diagram of the setup in Figure 4.11, the coupled tank consists of two uniform cross-sectional area tanks separated by a baffle, and there are individual level sensors and water pumps for each tank. By closing the baffle between the two tanks and utilizing only one tank, a liquid level plant similar to the one used in the simulations is obtained. This allows for the verification of the simulation results using a real world problem where disturbances and unmodelled plant dynamics cannot be avoided.

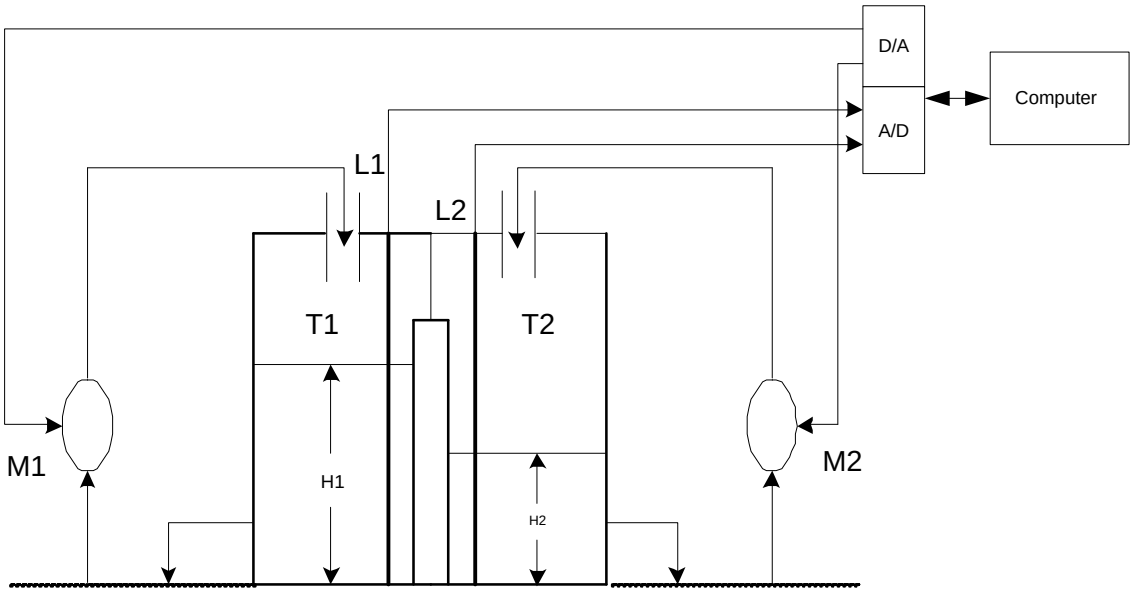

Figure 4.11. Schematic diagram of the Plant

Before actual control can take place, the plant must first be characterized. To start off, a noise analysis of the plant is performed by obtaining some samples and examining their frequency characteristics when the pump voltage is held constant at 2.3V. From Figure 4.12, the Fast Fourier Transform (FFT) of the voltage obtained from the level sensor reveals certain information about the plant. It is seen that the input voltage from the level sensor is subjected to the interference

of the AC power interference at 50Hz. Furthermore, there is a huge spike around 120Hz, probably due to the rotation frequency of the blades in the water pump. To effectively filter out both disturbances, over-sampling at 100Hz is chosen. Since the sampling rate is a multiple of the AC supply, the interference from the AC supply can be eliminated by taking block averages. It is also sufficiently small to reject noise from the water pump. Another visual observation through this open loop noise test is that the pump does not provide a constant flow rate when the command voltage is held constant. This causes the liquid level to drift around randomly. Due to the difficulty in filtering out this low frequency noise, the adaptive control scheme is given the job of overcoming this effect.

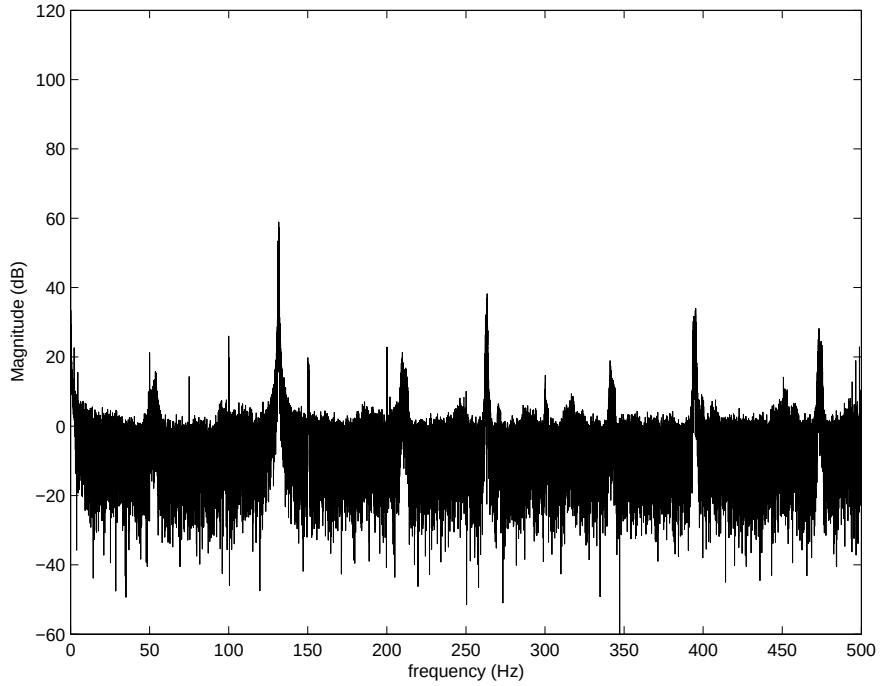

Figure 4.12. Noise analysis of the Liquid Level Plant

Next, the level sensor used is calibrated by noting the actual height of the liquid level and the voltage output of the level sensor. From the input voltage (V_{in}) verses the measured height (h_L) plot as shown in Figure 4.13, a simple linear relationship

between the two can be estimated to be

$$h_L = MV_{in} + C \tag{4.10}$$

where M and C are found to be 5.0938 and -1.9933 respectively.

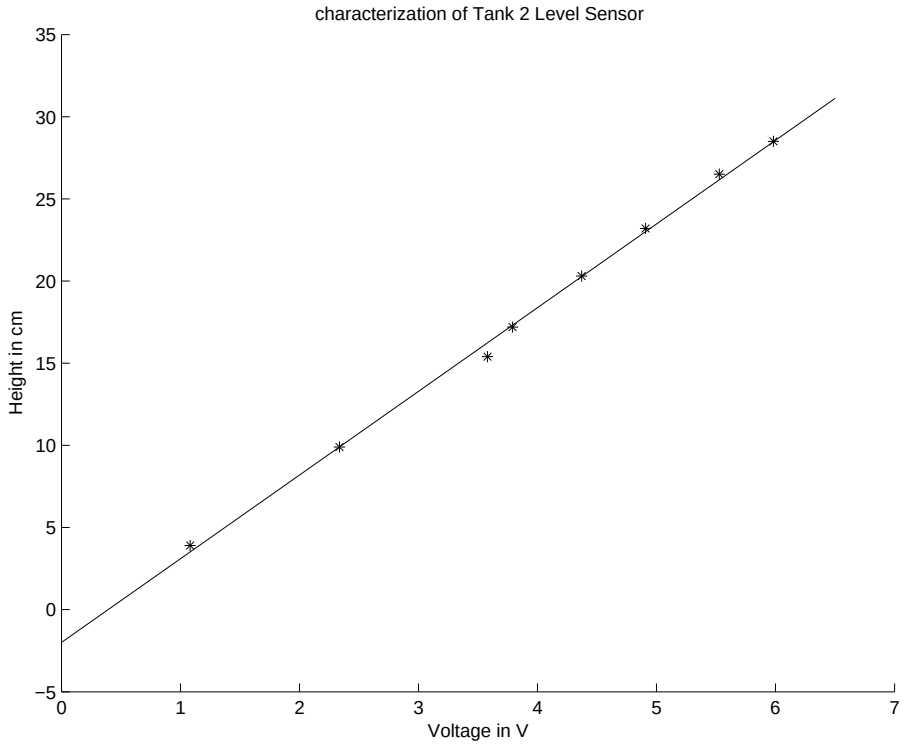

Figure 4.13. Level Sensor Characterization

Since the water flow rate, F_i, into the tank is controlled by manipulating the signal voltage sent to the pump, the relationship between the flow rate and the pump voltage is obtained by measuring the time it takes to fill up 1 liter of water. Figure 4.19 shows the relationship between the output control voltage and the flow rate can be estimated to be

$$F_i = M_q V_{out} + C_q \tag{4.11}$$

where M_q and C_q are 0.0627 and 1.1443 respectively.

Lastly, an idea of the liquid level process parameters is needed in order to commission the controller. To reduce the number of parameters to be estimated,

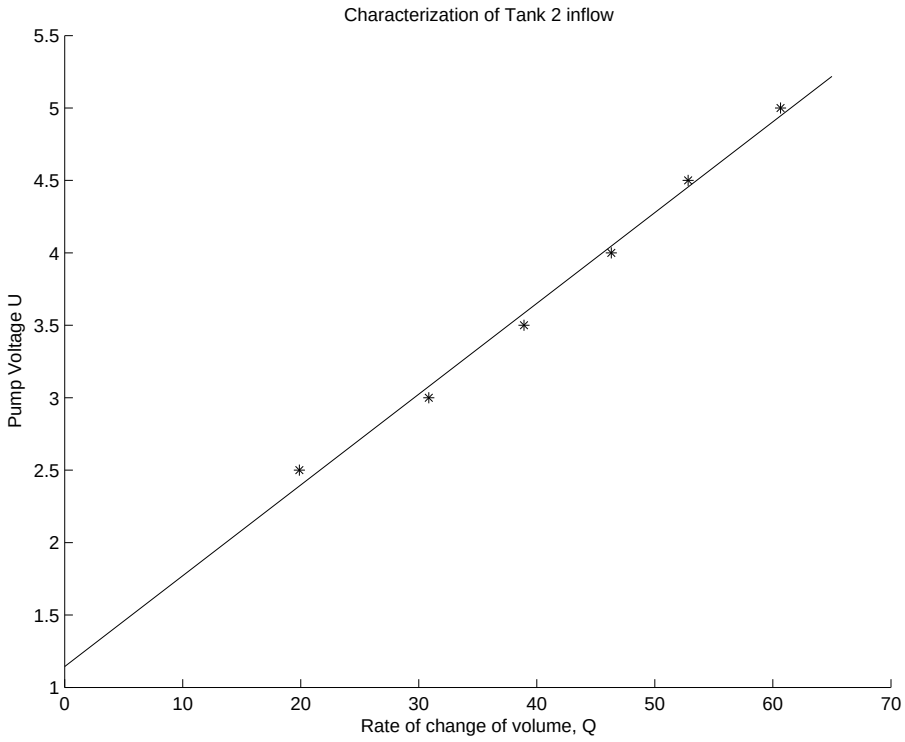

Figure 4.14. Characterization of Pump Flow rate

the differential equation describing the liquid level process (Equation (4.1)) is re-expressed as

$$A\frac{dh_L}{dt} = -\alpha\sqrt{h_L} + F_i \tag{4.12}$$

where A is the cross sectional area of the tank and α is the discharge coefficient. When there is no inflow of water into the tank, or $F_i = 0$, integrating Equation (4.12) from 0 to T seconds reveals

$$\frac{\alpha T}{A} = 2\left[\sqrt{h_L(0)} - \sqrt{h_L(T)}\right] \tag{4.13}$$

The discharge coefficient, α, can be measured experimentally by noting the amount of time it takes for a change in height of the water level and substituting them into Equation (4.13). With the cross-sectional area of the tank, A, measured to be $36.52cm^2$, α is estimated to be $5.6186scm^{1.5}$. This completes the description of the liquid level process to be controlled. Using these information, the next section will concentrate of the design of the self-learning control scheme.

4.5.2 Design of Controller

The control objective for the liquid level plant is to alternate the water level between 15, 20 and 25 cm at 5 minutes interval. The reference trajectory is generated by a first order model with time constant of 45 seconds to prevent excessive stress on the water pumps. Using the rule of thumb that the sampling time should be around 0.1-0.5 of the fastest dynamics present in the system (Astrom and Wittenmark, 1995), the sampling time is chosen to be 5 seconds.

Like the simulations presented earlier, the inputs to the neurofuzzy controller are the reference signal and its derivative. Three triangular fuzzy sets, centered at 15, 20, and 25cm respectively, are used to characterize the input space of the reference signal, $r(t+1)$, while the input space for $\Delta r(t+1)$ is partitioned by three triangular fuzzy sets with apexes at -0.6, 0 and 0.6.

To determine the PI parameters that could be used together with the commissioning strategy to select the controller parameters, relay auto-tuning for the experimental plant is done, and results are displayed in Figure 4.15. With the output relay amplitude set at $15cm^3s^{-1}$, and a deadzone of $0.1cm$ to counter the effects of noise, the plant oscillates with an amplitude of 2.57 cm with a period of 25 seconds. This translates to a proportional gain of 7.42 and an integral time of 20 seconds for a PI controller with the Zigeler-Nichols tuning rules. Using Equation (2.43), the parameters for the self-learning control scheme are 0.6364, 0.1054, 28.1507, and 1 for k_p, γ, λ_1, and δ respectively. With the weights set to zero initially, Figure 4.16 demonstrates that the neurofuzzy control scheme is able to learn to provide good control. In the actual experiment, the weights used are preinitialized to the values generated by the simulation to speed up the learning process.

The experimental results using the control scheme are shown in Figures 4.17. The neurofuzzy control scheme adapts well to cope with the unmodelled plant dynamics and the disturbances, with good control performance being exhibited by the second training cycle. From the output control action plot in Figure 4.19, it is evident that the strong regulatory action is needed to cope with the drift in of

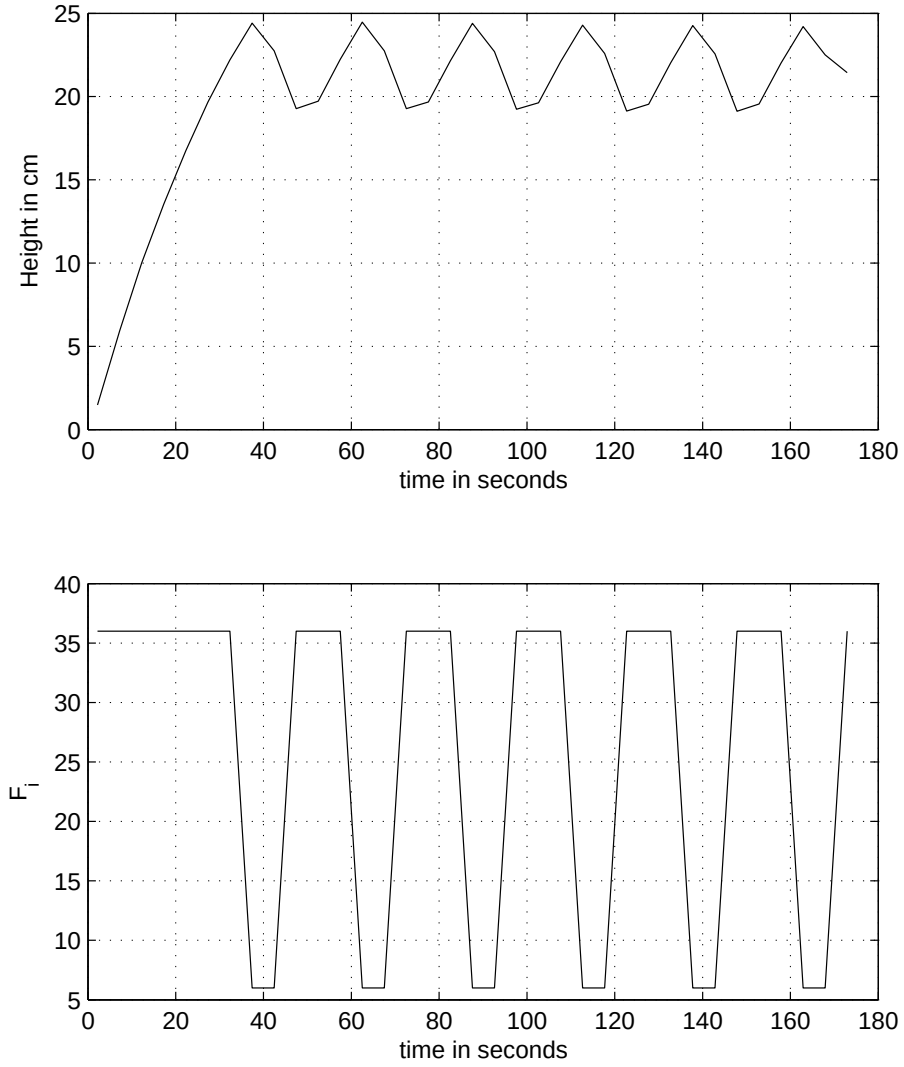

Figure 4.15. Relay auto-tuning results for the experimental liquid level plant

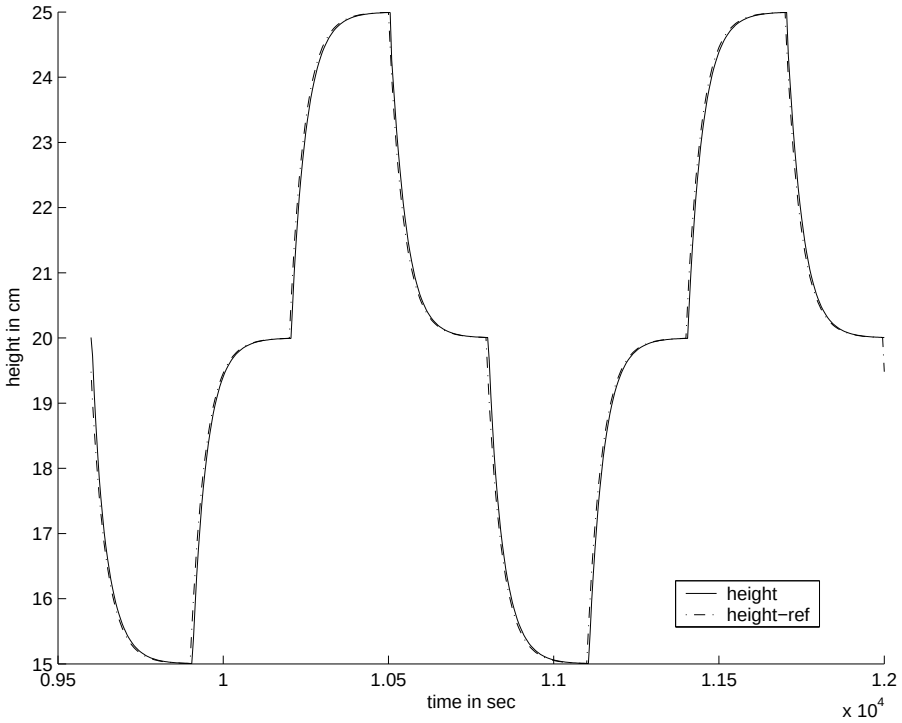

Figure 4.16. Simulated response of the coupled tank configured for liquid level control

the voltage needed by the water pump, with lower voltages required by the water pump to achieve the same liquid level. This also demonstrates the strong need for the proportional feedback controller in a practical system to reject such noise, and adaption of the weights of the neurofuzzy controller to achieve good control performance.

Overall, the neurofuzzy control scheme using the proposed FELS and NLMS algorithm has been successfully applied to a practical plant.

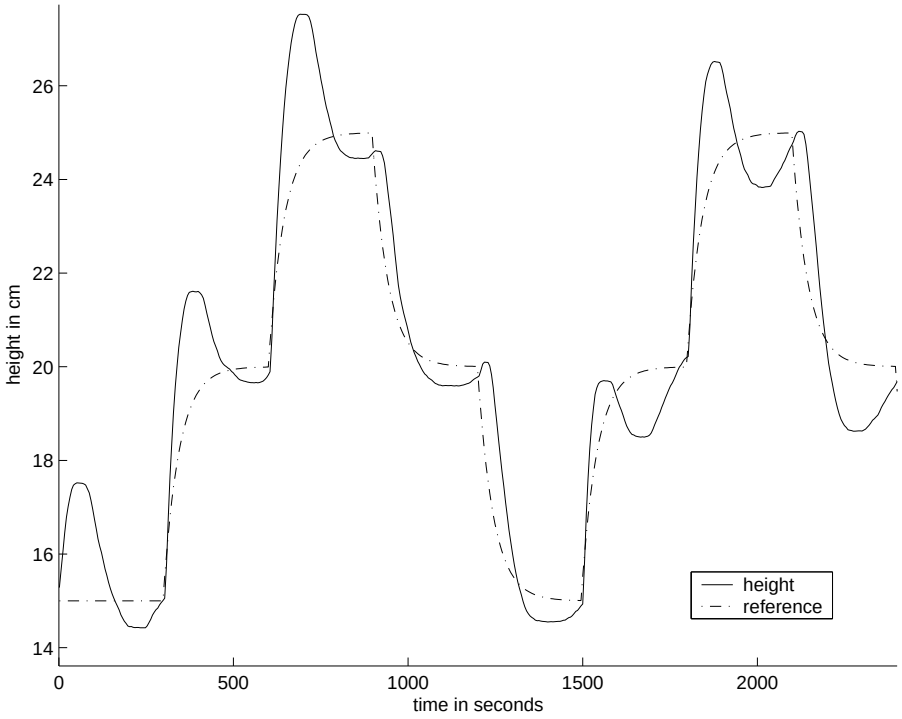

Figure 4.17. Initial control response of the liquid level plant

4.6 Conclusion

This chapter successfully analyzed the performance of the proposed feedback error learning strategy using a liquid level process. The rate at which the proposed feedback error learning strategy learns is shown to be superior to that of the original expression. An alternative commissioning strategy for the proposed FELS is found to offer tuning parameters with similar control performance as the generic

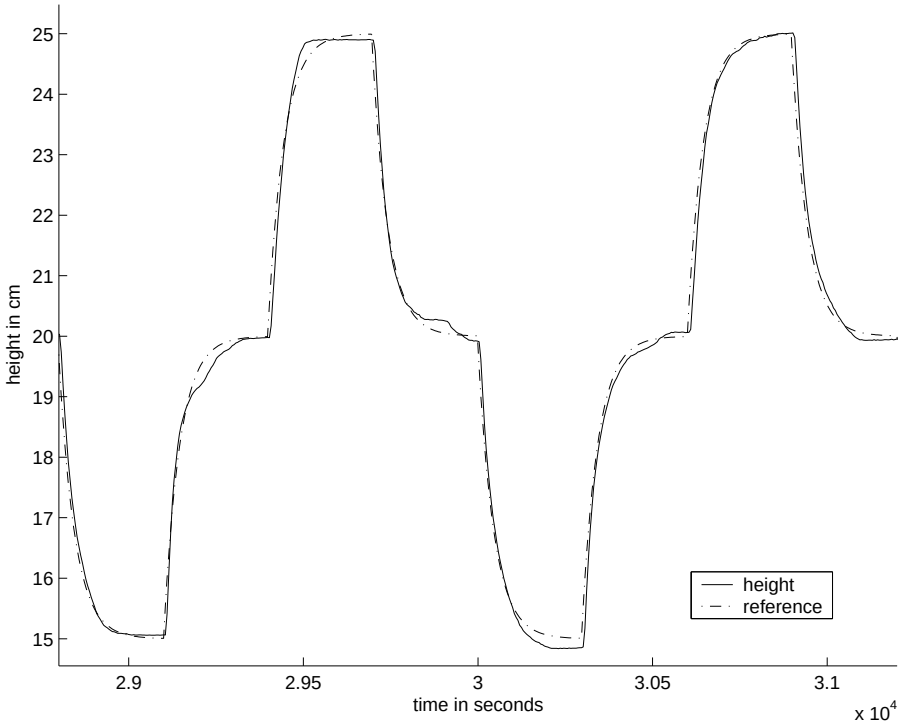

Figure 4.18. Experimental control performance after training

commissioning strategy. This suggests that there is a duality in the choice of the commissioning strategy for the proposed FELS. Experiments carried out on a actual plant confirms the feasibility of the proposed strategy in practice.

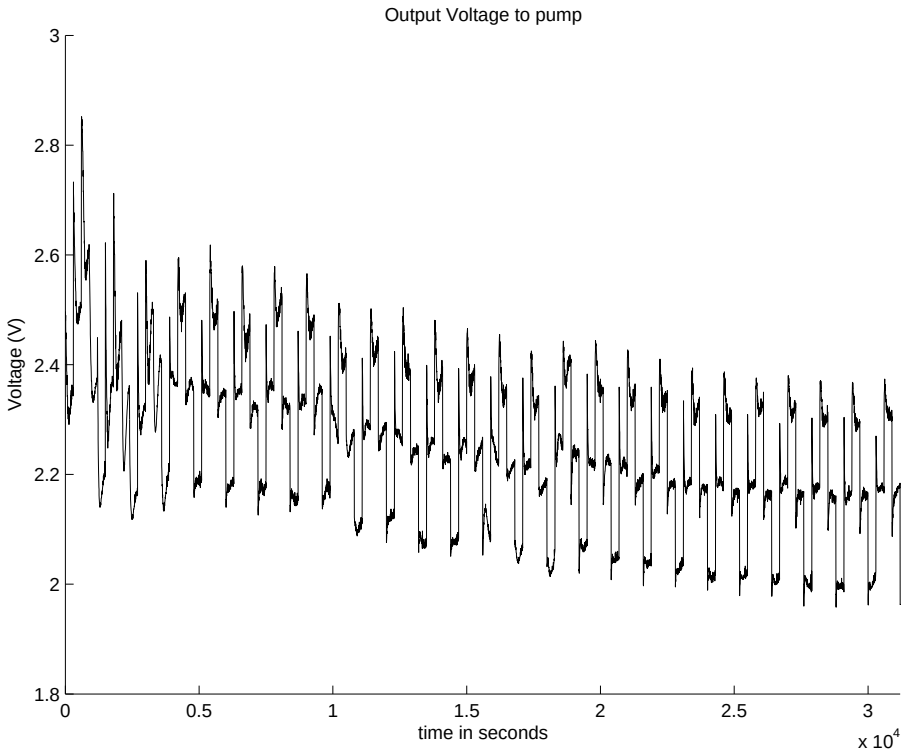

Figure 4.19. Output voltage to the pump in experiment

Chapter 5

Neurofuzzy pH Control

5.1 Introduction

The control of pH is a critical factor in many biological and industrial processes, as poor pH control can lead to inferior products and/ or detrimental consequences. Despite extensive research, pH control remains a challenging problem for practicing control engineers and researchers. The difficulty with pH control lies in its inherent severely nonlinear titration relationship, which exhibits extreme sensitivity around the equivalence point and relative indifference to control efforts at the ends of the pH curve. Furthermore, changes to the chemical composition in the process stream will result in a time varying process that further complicates the control problem.

Numerous control schemes have been employed on this sensitive nonlinear problem. In recent years, various neural networks and fuzzy logic techniques have also been used on this control problem (Abonyi *et al.*, 2001). While some of these schemes have treated the nonlinear process as a black box, others have incorporated *a prior* structural information about the pH process into their control strategies in order to ease and speed up the learning process.

One problem faced in the incorporation of *a prior* information to achieve "more" intelligent control is that the knowledge embedded may be rendered inaccurate in the face of changing dynamics in pH control. This is especially relevant in some pH control applications like the treatment of waste water as the exact composition

of the ions in the process is unknown. When unknown buffers are introduced into the process stream, the resultant plant dynamics can differ significantly from its nominal one.

Even without the addition of buffers, changing flow rates and/ or concentration will cause the pH plant dynamics to be time varying. This makes on-line adaption of the pH controller to enable it to cope with the variations almost a prerequisite for good control performance. Yet, in many adaptive controllers, information about the process structure needs to be known. The uncertainty in the pH dynamics thus poses a severe test on capability of any adaptive controller to cope when there are mismatches between the plant and the model.

In this chapter, a study on the feasibility of using the neurofuzzy control scheme to control a pH neutralization process in a Continuously Stirred Tank Reactor (CSTR) is attempted. Instead of evaluating the optimization algorithms used in the neurofuzzy control scheme, the focus is changed to studying the effects of incorporating *a prior* structural information on control performance. Comparisons are made to discover whether there is merit in using *a prior* knowledge that may be inaccurate due to the rapidly changing buffering conditions.

The organization of this chapter is as follows : First, the mechanics of the pH process in a CSTR is described to provide background information. It is shown that certain types of pH process may be approximated as a Wiener model, which consist of a linear dynamic part and static nonlinear part. A method for incorporating this structural knowledge into the control scheme is described. Due to problems in coping with buffering changes, adaptation of the structural knowledge is carried out in the next section to see if it leads to improved performance when there are changes to buffering. The generic neurofuzzy control scheme, which does not utilize any structural information, is presented next, and comparisons between the structures' performance are made. Section 5.4.2 then presents the experimental results obtained using a pilot pH plant to verify the feasibility of the neurofuzzy control scheme. Finally, conclusions are made.

5.2 The pH plant

In this section, a review of the pH process that is to be controlled is given. First, the static pH neutralization process is derived from first principles involving the ion balances and chemical equilibrium relations. The difficulty involved in the control of pH is also explained. From this understanding of the static pH process, the dynamic pH process in a Continuously Stirred Tank Reactor (CSTR) is then described.

5.2.1 The static pH process

The reagents in a pH neutralization reaction act either as acids or bases. Acids and bases are substances that are capable of either donating or accepting hydrogen ions, such that acids are proton donors while bases are proton acceptors. Protons can also be denoted as hydrogen ions, and its concentration is a measurement of the level of acidity of a substance. The pH variable is thus defined as

$$pH = -\log_{10}[H^+] \tag{5.1}$$

where $p(\cdot)$ is the operator denoting *taking the negative logarithm to the base 10* and $[\cdot]$ is the concentration of the respective ion.

The strength of an reagent is classified according to the fraction of molecules that dissociate, or the degree of dissociation α. An strong reagent is one which dissociates completely in water, while a weak one only partially dissociates. Consider a weak monoprotic acid, HA, where A^- is the anion. An equilibrium is set up between undissociated molecules HA and the ions H^+ and A^- in water :

$$HA \rightleftharpoons H^+ + A^- \tag{5.2}$$

At equilibrium, the acid dissociation constant, K_a, for Equation (5.2) is given by

$$K_a = \frac{[H^+][A^-]}{[HA]} \tag{5.3a}$$

$$pK_a = -\log_{10} K_a \tag{5.3b}$$

where the square brackets represent concentration in $moldm^{-3}$. The case is analogous for bases when they dissociate to form hydroxide (OH^-) ions. A reagent's strength increases when their pK value decreases. Reagents with negative pK are called strong while those with pK greater than one are considered weak. When the reagent is polyprotic, or that the reagent is capable of dissociating more than one hydrogen or hydroxyl ion per molecule, the reagent will have a dissociation constant for each hydrogen or hydroxyl ion.

In the special case for water at 25^oC, the product of the hydrogen ions and the hydroxide ions is equal to $10^{-14}mol^2dm^{-6}$. This product is known as the ionic product of water K_w, i.e.

$$K_w = [H^+][OH^-] = 10^{-14} \tag{5.4a}$$

$$pK_w = 14 \tag{5.4b}$$

The pH titration relation is a mapping of the input titrants to the output pH measurement. It is a static relationship and is derived from the need for ionic equilibrium in the reaction. As an example, consider the neutralization reaction of a strong acid like hydrochloric acid (HCl) and a strong base like sodium hydroxide ($NaOH$) :

$$HCl + NaOH \rightarrow NaCl + H_2O \tag{5.5}$$

Using the electro-neutrality condition, the nett sum of the ionic charges must be zero, i.e.

$$[Na^+] + [H^+] = [Cl^-] + [OH^-] \tag{5.6}$$

Denoting $x_a = [Cl^-]$ as the acid ionic concentration and $x_b = [Na^+]$ as the base ionic concentration, the neutralization equation for the titration process can be rewritten as

$$x_b = x_a - 10^{-pH} + 10^{pH-14} \tag{5.7}$$

Equation (5.7) is the titration relation for the reaction between HCl and $NaOH$, and is graphically illustrated in Figure 5.1. As seen from the figure, the titration curve is a 'S' shaped curve, with most of the pH curve relatively flat except for the portion around the equivalence point. The extremely steep gradient

around the equivalence point causes the pH process to be extremely sensitive to variations. This characteristic is the reason why pH control is very difficult. The rangeability of the control valves need to be immensely great to effectively bring about the required change in pH value. On the other hand, the regions at the two ends of the pH titration curve is relatively insensitive to changes in the ionic concentration. The extreme variation in the gain of the static pH process raises difficult stability and performance issues.

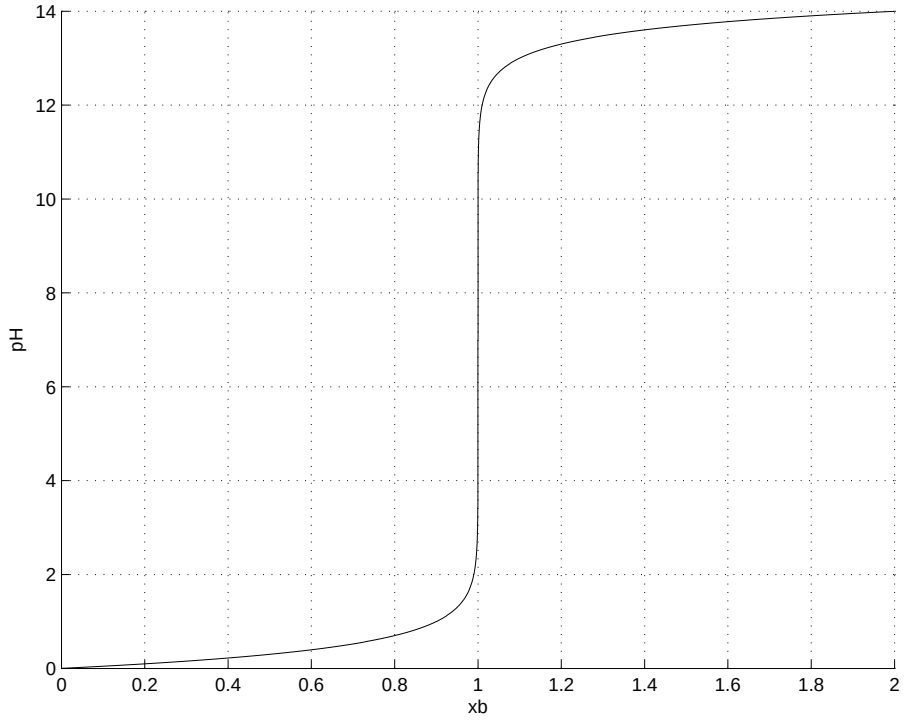

Figure 5.1. Titration curve for a strong acid, strong base reaction

The severe process nonlinearity is further complicated by the effects of buffering when weak reagents are used. A weak reagent acts as a buffer when it resists changes in pH. To elaborate, consider a weak acid acetic acid (CH_3COOH) reacting with a strong base $NaOH$ to produce the following equilibrium :

$$CH_3COOH + NaOH \rightleftharpoons NaCH_3COOH + H_2O \tag{5.8}$$

As the ionic charges must balance, the following equation should hold :

$$[Na^+] + [H^+] = [CH_3COO^-] + [OH^-] \tag{5.9}$$

Since the acid dissociation constant, K_a, of acetic acid is

$$\begin{aligned} K_a &= \frac{[CH_3COO^-][H^+]}{[CH_3COOH]} \\ &= 10^{-4.75} \end{aligned} \tag{5.10a}$$

and expressing $x_b = [Na^+]$ as the ionic base concentration, $x_a = [CH_3COO^-] + [CH_3COOH]$ as the ionic acid concentration, the following titration relation can be derived :

$$x_b = \frac{x_a}{1 + 10^{4.75-pH}} - 10^{-pH} + 10^{pH-14} \tag{5.11}$$

Comparing Equation (5.11) with Equation (5.7), it may be seen that the difference in the titration relationship involving weak acid and strong acid is the extra $\frac{1}{1+10^{pK_a-pH}}$ term. The impact of this additional factor is graphically shown in Figure 5.2. It is seen that multiplying the additional multiplication term to x_a causes a inflection point on the process titration curve at the reagent's pK value, or that there is resistance to changes in the pH value when compared to its strong acid counterpart. This resistance to pH changes is known as the buffering effect, and can be explained as follows. In the buffering region, the concentration of the CH_3COO^- ion and the CH_3COOH acid molecule is very large compared to the H^+ ion. When additional H^+ ions enter the buffered solution, they will react with $CH3_COO^-$ to form CH_3COOH, while the entry of OH^- ions on the other hand will cause CH_3COOH to ionize to replenish the H^+ ions in the solution. Thus, changes in pH are reduced.

The equation for a general pH titration relationship with M reagents is as follows (Wright, 1998) :

$$\sum_{i}^{M} s_i(pH) = 10^{-pH} - 10^{pH-14} \tag{5.12}$$

where $s_i(pH)$ is a function of pH and the dissociation constants for the i^{th} species. Table 5.2.1 illustrates the contribution to the titration relationship for some types of acids and bases.

Having gained an understanding on the static pH titration, the next step is to see how the static pH titration is related to the actual dynamical process described in the next section.

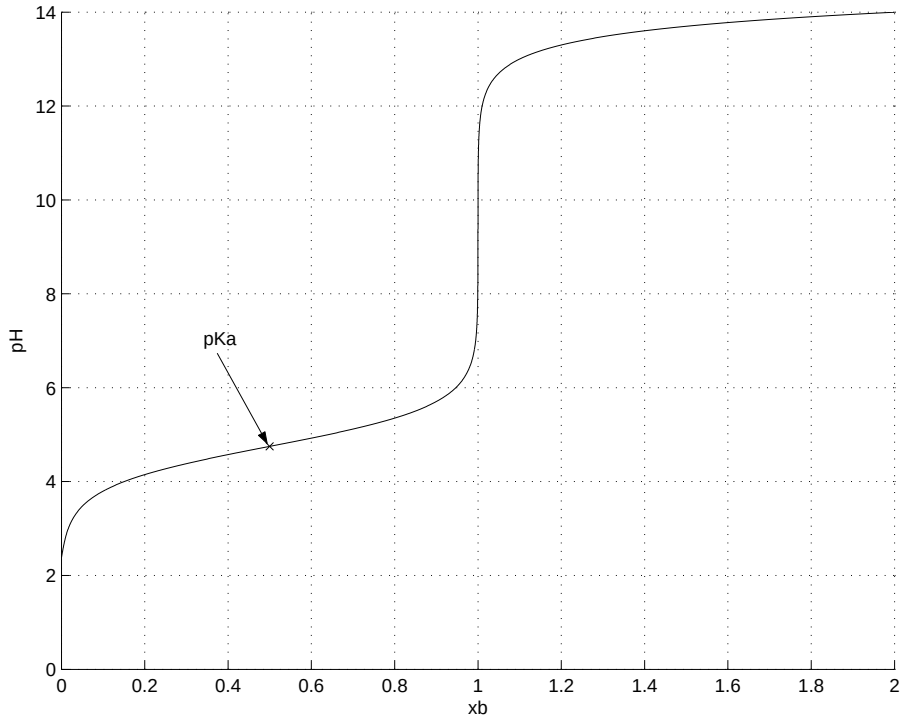

Figure 5.2. Titration curve for a weak acid, strong base reaction

i^{th} ionic species	ionic concentration	$s_i(pH)$
anion,HA	$[HA]+[A^-]$	$-\frac{1}{1+10^{pK_a-pH}}$
anion,H_2A	$[H_2A]+[H_A^-]+[A^{2-}]$	$-\frac{2+10^{pK_{a2}-pH}}{1++10^{pK_{a2}-pH}+10^{pK_{a1}+pK_{a2}-2pH}}$
cation,BOH	$[BOH]+[B^+]$	$\frac{10^{-pH}}{10^{-pH}+10^{pK_b-pK_w}}$
cation,$B(OH)_2$	$[B(OH)_2]+[BOH^+]+[B^{2+}]$	$\frac{2x10^{-2pH}+10^{pK_{b2}-pK_w-pH}}{10^{-2pH}+10^{pK_{b2}-pK_w-pH}+10^{pK_{b1}+pK_w-2pK_w}}$

Table 5.1. Definitions of $s_i(pH)$

5.2.2 pH process in a CSTR

The pH neutralization process considered in this thesis is assumed to take place in a Continuously Stirred Tank Reactor (CSTR), as shown in Figure 5.3. It consists of an influent acid as the process stream, an influent base as the titrating stream, and an effluent stream to maintain the solution volume in the tank as a constant.

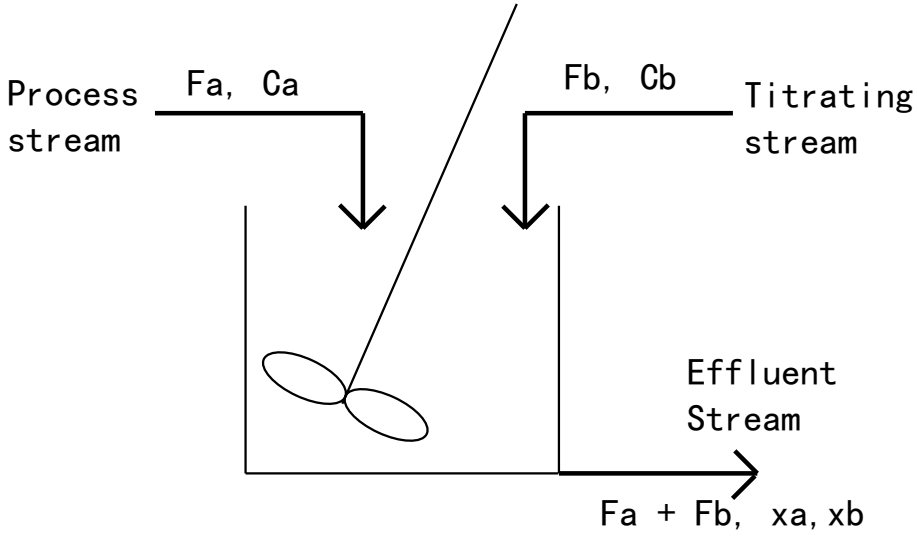

Figure 5.3. The CSTR configuration

This environment has several assumptions : (i) the volume of the solution in the tank is a constant, (ii) the solution is mixed perfectly, (iii) the chemical reactions remain isothermal, and (iv) the chemical reactions are assumed to attain chemical equilibria instantaneously.

The dynamics for the mixing process as described in McAvoy *et al.* (1972) are

$$\frac{dx_a}{dt} = F_a C_a - F_T x_a \tag{5.13a}$$

$$\frac{dx_b}{dt} = F_b C_b - F_T x_b \tag{5.13b}$$

with subscripts a and b representing the acid and base species. F_i (liter/min), C_i (mol/liter), and x_i (mol/liter) are the flow rate, concentration, and ionic concentration of the i^{th} species respectively. $F_T = F_a + F_b$ is the sum of the flow rates, and V is the mixture volume in liters. The mixing dynamics defined in Equation (5.13) are derived based on the principle of mass conservation of the individual ionic

components (Stephanopoulos, 1984), with the assumptions of having a constant volume and perfect mixing. By assuming that the effluent flow rate F_T is linearly related to the hydrostatic pressure of the tank liquid level thorough the effluent outlet resistance (Stephanopoulos, 1984), the time constant of the tank can be expressed as

$$\tau = \frac{V}{F_T} \tag{5.14}$$

The mathematical model for the pH process in a CSTR is a combination of the CSTR dynamics and the static titration equation. First, the mixing dynamics of the CSTR in Equation (5.13) gives the ionic concentrations from the influent flow rates and concentrations of the solutions involved in the process. Substituting these ionic concentrations into the static titration relation in Equation (5.12) and solving it, the pH value can then be obtained.

From Equation (5.13), it can be seen that the CSTR dynamics is only mildly nonlinear. In practice, the concentration of the reagent in the titrating stream can be chosen such that $F_a >> F_b$ hold. This implies that the bilinear dynamics can be approximated as linear dynamics :

$$\frac{dx_a}{dt} \approx F_a(C_a - x_a) \tag{5.15a}$$

$$\frac{dx_b}{dt} \approx F_b C_b - F_a x_b \tag{5.15b}$$

To sum up, the equations indicate that the process of neutralizing a weak acid with a strong base in a CSTR may be approximated by the Wiener type non-linear model shown in Figure (5.4). The linear block is the approximate CSTR dynamics defined by Equation (5.15), while the static nonlinear function is the titration curve (Equation (5.12)).

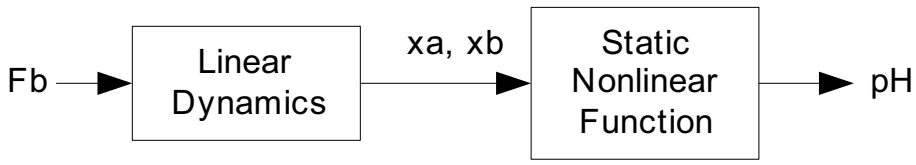

Figure 5.4. The Wiener nonlinear model

A common method for controlling Wiener models is to employ a static inverse model to cancel the titration nonlinearity before dealing with the CSTR dynamics. However, the performance of the Wiener-model control strategy often hinges on how well the nonlinearity is cancelled. When the neutralization curve changes drastically due to buffering variations or flow rate changes, the inverse model may be inaccurate and the performance of the control scheme may suffer. In the next section, the usefulness of using this *a prior* information for pH control will be analyzed.

5.3 Simulation and Analysis

5.3.1 Simulation setup

The simulations that are performed to investigate the effects of structural differences on pH control performance are detailed here. The pH neutralization process that is considered mixes a weak ethanoic acid (CH_3COOH) of 0.05N with a strong base ($NaOH$) of 0.1N. The acid flow rate is kept constant at 400 ml/min, and the base is used as the titrating reagent to manipulate the pH level.

The study is conducted by using the control schemes to track transitions between different pH levels. The set point changes are smoothen by a first order reference model with a time constant of 20 seconds. Two tests were performed. The first test examines their performance under nominal conditions. In the second test, carbonic acid (H_2CO_3), a diprotic reagent with pKa of 6.35 and 10.25 at 0.2N for the first and second hydrogen ion, is added to the pH plant according to the schedule shown in Table 5.2. In order to understand the changes that are brought about by the introduction of the buffer, the titration curves are shown in Figure 5.5. The plots clearly show that the buffer has a significant impact on the shape of the titration curve at high pH levels. This is because the normality of the buffer used is quadruple that of the acetic acid. The nett result is that the quality of any *a prior* information used that is incorporated in the controller will degrade severely. To prevent the control performance from deteriorating, the controller has

to learn to adapt to the prevailing neutralization characteristics on-line.

Buffer flowrate, F_c (ml/min)	Time (min)
0	0-600 and 2400-3000
100	600-1200 and 1800-2400
200	1200-1800

Table 5.2. Buffer flowrate variation schedule

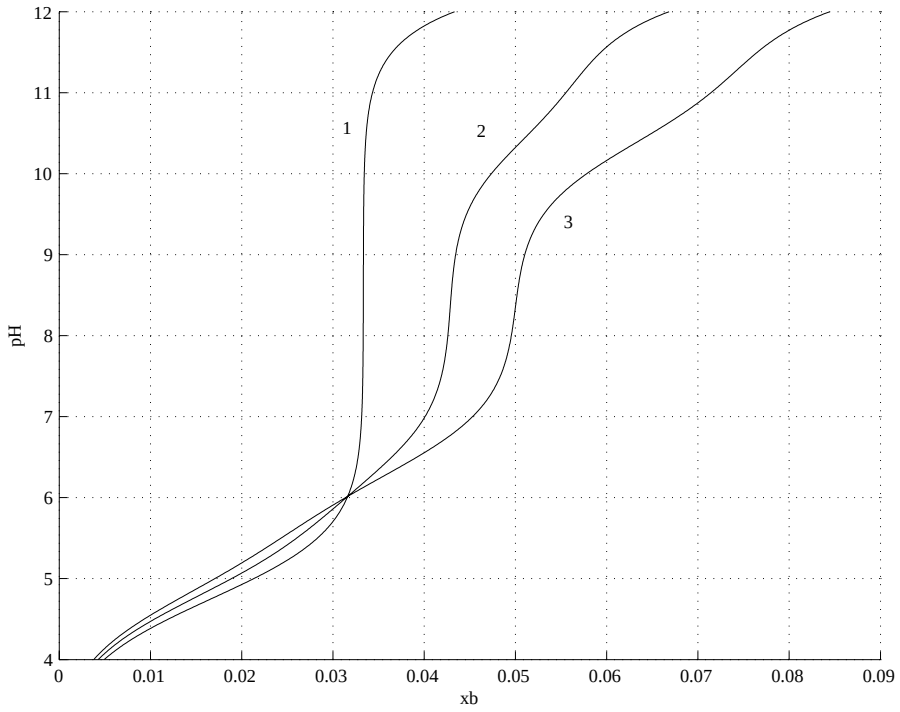

Figure 5.5. Titration relationship between xb and pH
under different buffer flow rates, F_c
(1) F_c=0, (2) F_c=100, (3) F_c=200 ml/min

5.3.2 Wiener-model controller

To begin the analysis, the structural information presented in Section 5.2.2 is used to derive the controller architecture shown in Figure 5.6. The static inverse titration model, h^{-1}, transforms the output pH value into an estimate of the base ionic concentration (x_b^*). pH control is then performed in this space using the

adaptive neurofuzzy controller, which then provides the required titrating flow rate, F_b. It is expected that learning the inverse of the transformed $x_b - F_b$ space should be easier, as the relationship is approximately linear (Equation 5.15).

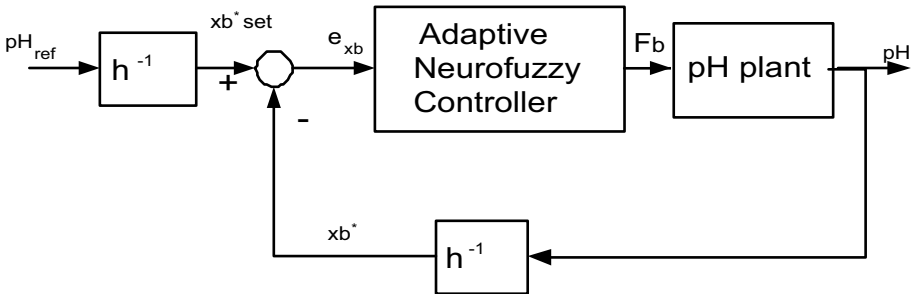

Figure 5.6. Structure of the Wiener-model controller

A B-spline network, with 13 evenly spaced second order B-splines spanning the input pH domain from 5 to 12, was used to model the inverse neutralization relationship. Input output data obtained from the nominal acid-base titration curve was used to identify the network parameters off-line. Figure 5.7 shows the percentage error in modelling the titration relationship. It is seen that the modelling error around the more sensitive regions (pH 7 to 10) are small, and moderate for the other regions. This is important, as errors made in the sensitive region will be amplified significantly. The resulting average modelling error (difference between the inverse model and the actual titration curve) is 0.0018%.

As the inverse titration relationship h^{-1} is only an approximate relationship, zero error in the x_b space does not necessarily mean that the actual pH valve is equal to its reference. To ensure that the feedback error learning rule actively tries to minimize $(pH_{ref} - pH)$, the difference between the reference pH value and the actual pH is used to estimate the required feedforward control action i.e. Equation (2.13) becomes

$$\hat{u_f}(t) = u_f(t - t_d) + \gamma(pH_{ref} - pH) \tag{5.16}$$

The usage of the original FELS here, and also for the rest of the controllers in this discussion, is to simplify the learning, since the focus here is on the structural differences. The reference base ionic concentration, $x^*_{b,set}$, its rate of change

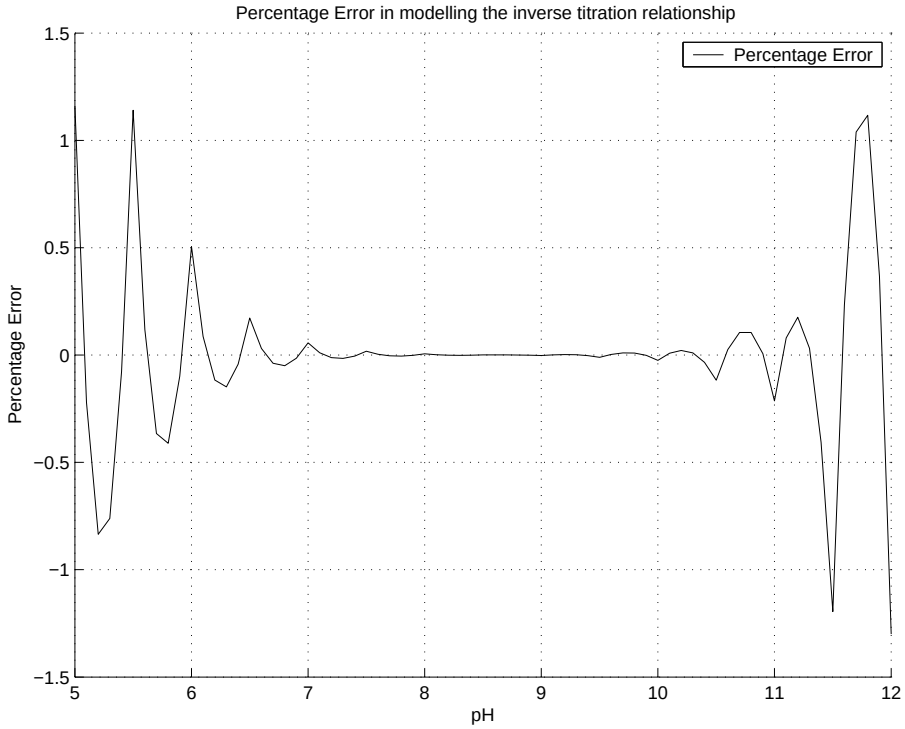

Figure 5.7. Percentage Error in modelling the inverse titration relationship, h^{-1}

($\Delta x^*_{b,set}$), and the base ionic concentration, x^*_b, derived from the static inverse model, are used as the inputs of the neurofuzzy controller. The input domains were spanned by 5, 2, 2 triangular fuzzy sets respectively. All the elements in the weight vector of the neurofuzzy controller are arbitrarily initialized to 0.1. In the selection of the parameters for the neurofuzzy control scheme, a problem faced is the difficulty in finding good PI/PID parameters that can be used in the commissioning strategy due to the sensitivity of the pH control problem. Therefore, manual tuning is used, and the parameters are as follows : γ is 0.03, k_p is 0.18, the NLMS algorithm's update rate (δ) is 1 and sampling time is 5 seconds.

Figure 5.8 shows the performance of the Wiener-model controller under nominal conditions. It shows that rapid convergence to the reference pH trajectory is obtained. While good tracking performance occurs in the mid pH range (7 to 11), the step responses at the bottom (6 to 7) and top (11 to 11.5) of the pH test range exhibit a mild overshoot and slightly sluggish behaviour respectively. Figure 5.9 displays the performance of the control scheme when the pH process is influenced

by carbonic acid according to the schedule shown in Table 5.2. The performance of the control scheme degenerates upon the introduction of the buffer. This effect is more prominent in the higher pH levels, probably because of the larger changes to the titration curve in this region (See Figure 5.5). The results indicate that the adaptive neurofuzzy controller is unable to compensate for the deviations in the inverse neutralization curve brought about by the addition of carbonic acid. It may be intuitively be argued that the use of erroneous *a prior* information caused the quality of the control to worsen. A natural extension is to try to modify the inverse titration model on-line.

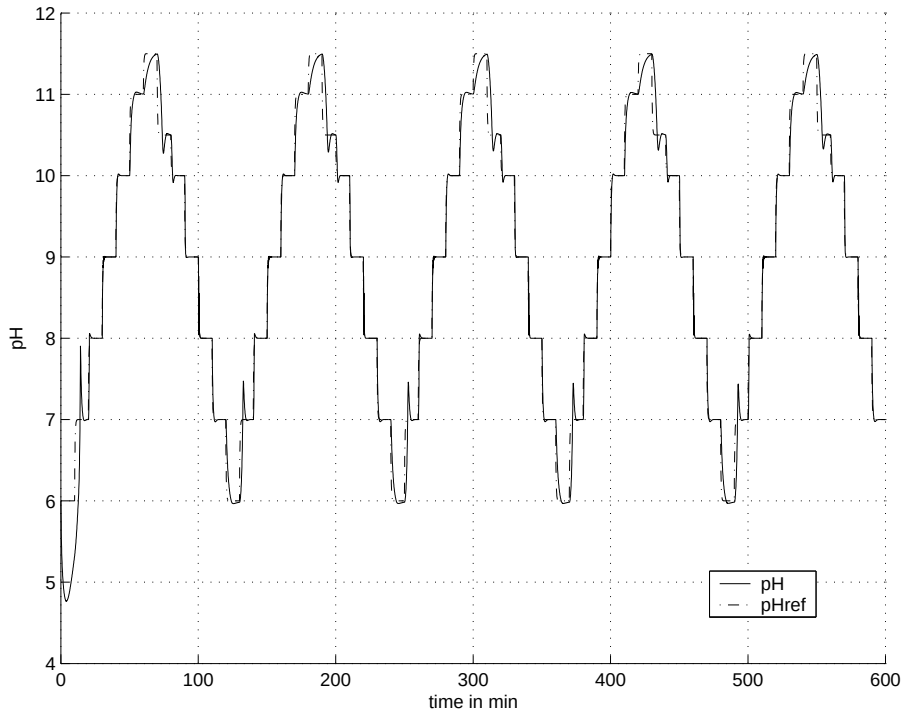

Figure 5.8. Performance of the Wiener-model controller under nominal conditions

5.3.3 Adaptive Wiener-model Controller

To adapt the inverse neutralization model on-line, the actual ionic concentrations of all the reagents must be known. This is almost impossible to achieve in practice for many pH control applications, due to the uncertainty of the reagents present

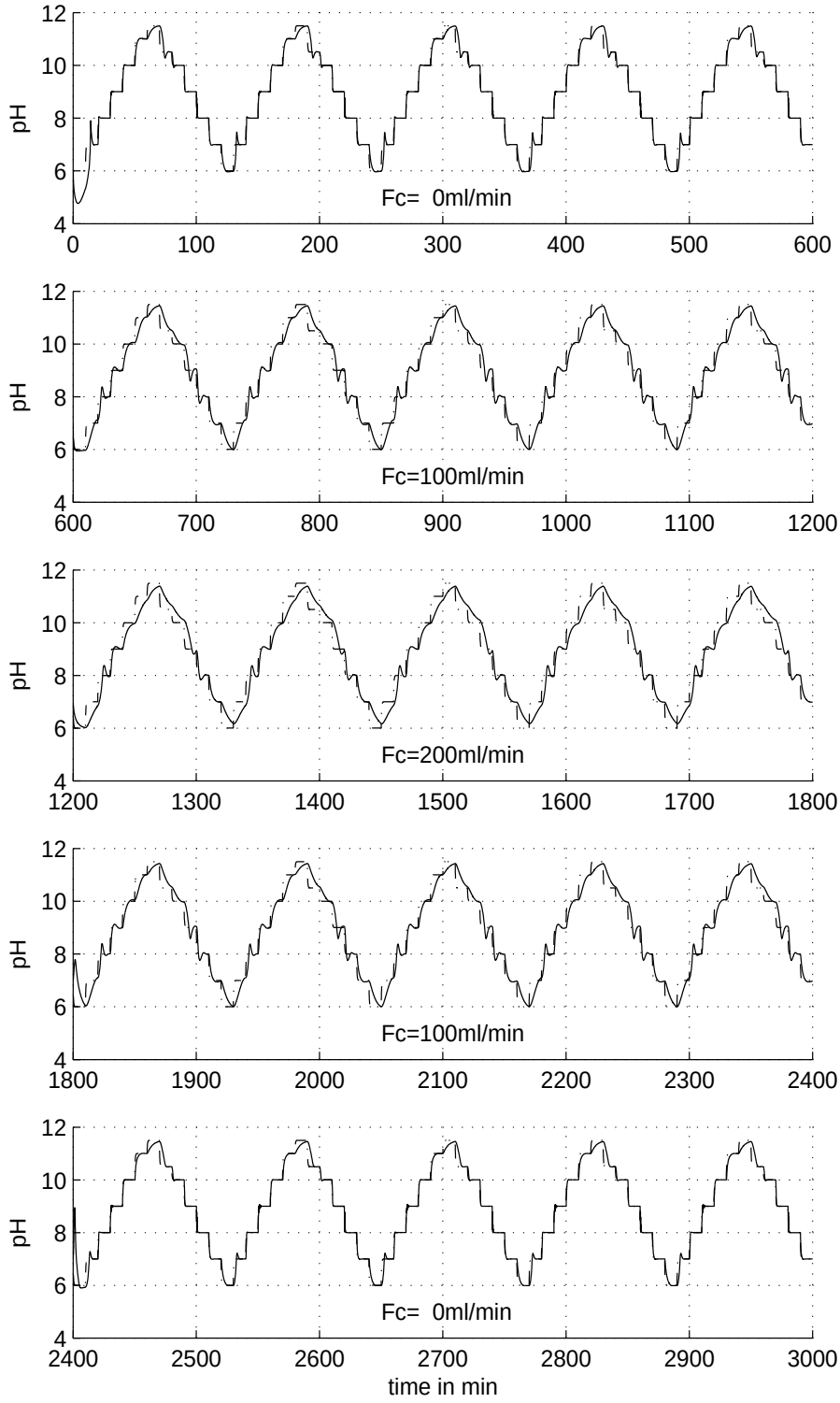

Figure 5.9. Performance of the Wiener-model controller under varying buffer flow rates

like in waste water treatment. One way to circumvent this problem is to used the theoretical relationship, as defined in Equation (5.15), to estimate the actual ionic concentration corresponding to a pH value.

At each sampling instant, the basic ionic concentration estimated using Equation (5.15), together with the measured pH value, are fed to a NLMS algorithm (learning rate, δ, is 0.75) to update the inverse titration relationship (h^{-1}). In order to provide a common basis for comparison, the structure of the neurofuzzy feedforward controller was not changed. The proportional gain of the conventional controller (k_p), the learning rate for the feedback error learning scheme (γ) and the sampling time were chosen as 0.1, 0.05 and 1 seconds respectively.

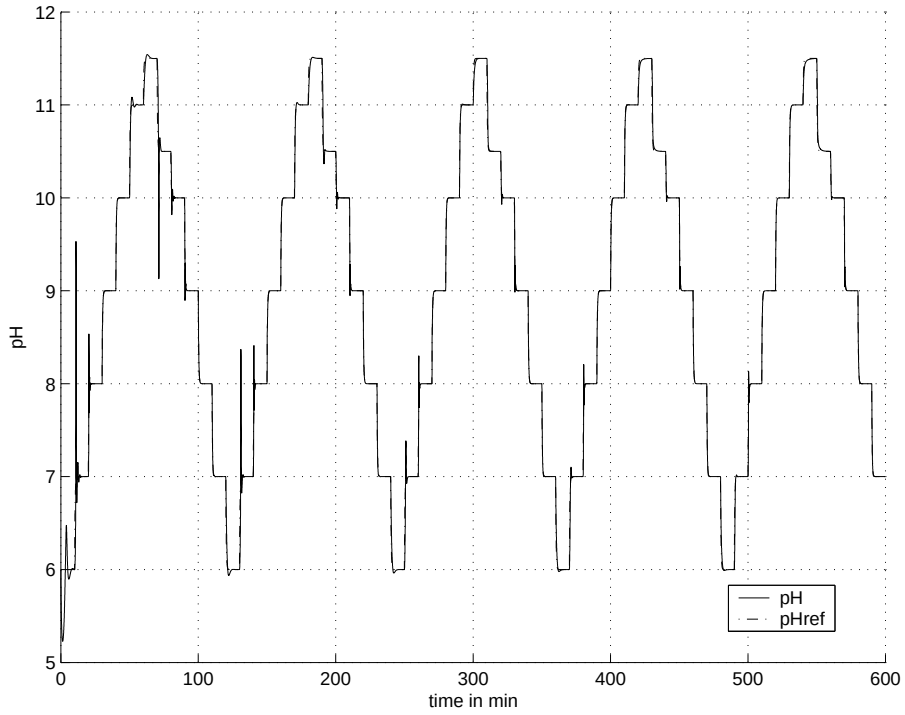

Figure 5.10. Performance of the adaptive Wiener-model controller under nominal conditions

The step responses obtained using the adaptive Wiener-model controller is shown in Figure 5.10. Compared with the results obtained using an inverse model that is not adapted on-line (Figure 5.8), better tracking performance was obtained. In particular, the step responses no longer overshoot the set point in the 6-7 pH

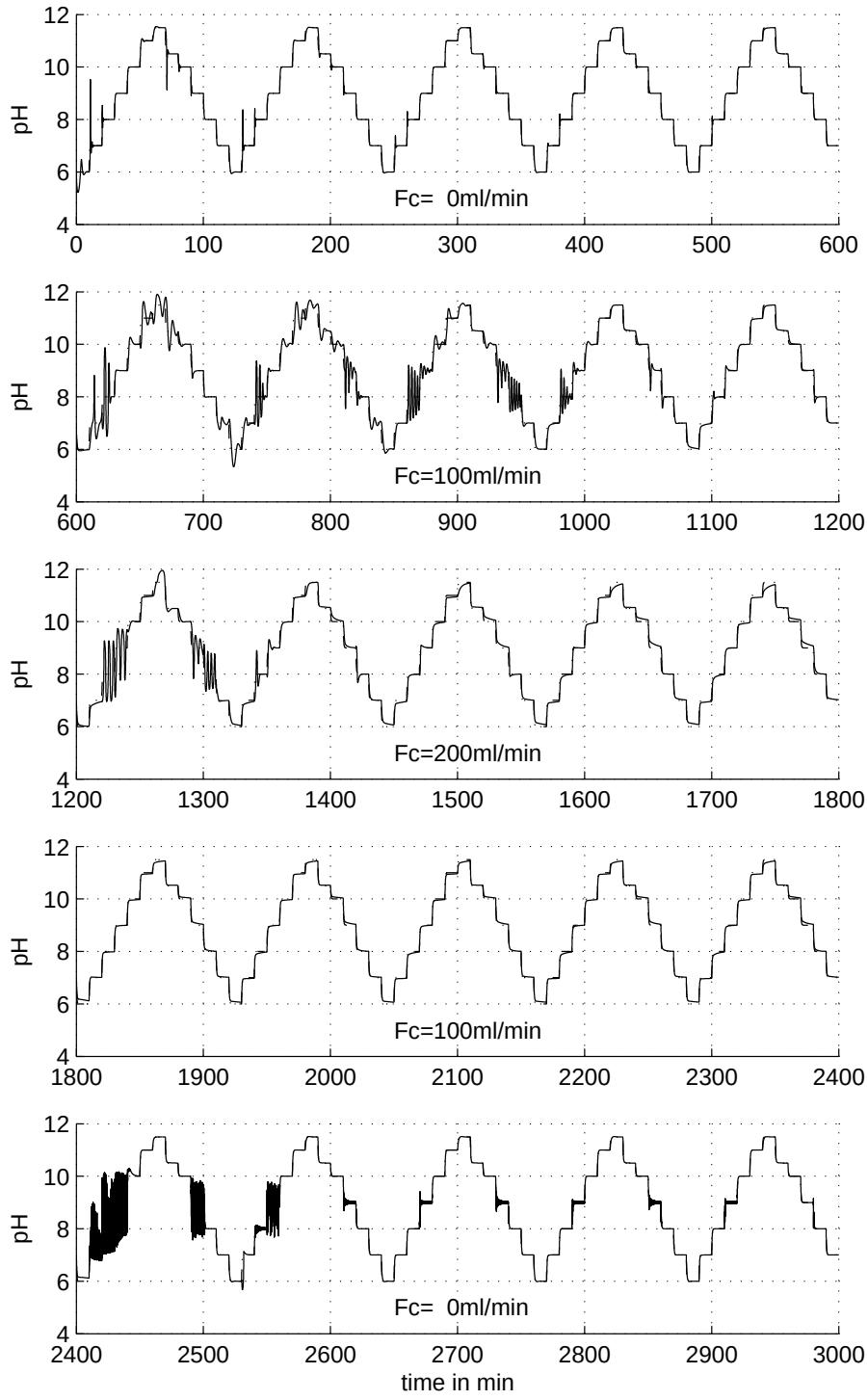

Figure 5.11. Performance of the adaptive Wiener-model controller when an unknown buffer is introduced

range while the speed of response is faster at high pH values. Figure 5.11 shows the control responses when the pH plant is subjected to variations in the amount of buffering. Although the adaptive Wiener-model controller is able to better reject the undesirable effects of the unknown buffer, the control performance is still not very satisfactory. Following a change in the flow rate of carbonic acid from 100 ml/min to 0 ml/min, the pH value oscillates wildly when the set point is in the sensitive pH region. As several training cycles are needed to eliminate the oscillations, the adaptive Wiener-model controller will be of limited use in practice.

5.3.4 Adaptive neurofuzzy control : a "Black Box" approach

Simulation results in the previous sections indicates that including the Wiener model representation in the control scheme may not be practical. Therefore, attention is turned to examining the feasibility of using the neurofuzzy control scheme without this structural information. The inputs to the neurofuzzy feedforward controller are selected as the reference pH level, $pH_{set}(k)$, its rate of change $\Delta pH_{set}(k)$, and the control action $U(k)$. Eight uniformly distributed triangular fuzzy sets partition the universe of discourse for the first input, $pH_{set}(k)$, while the input space for $\Delta pH_{set}(k)$ and $U(k)$ are partitioned by two fuzzy sets each. Due to the wide range of the pH set points, it is difficult for the adaptive neurofuzzy controller to learn at a appropriate rate using a common set of controller parameters. Hence, the controller parameters are scheduled according to the region in which the process is operating. When the reference pH levels are between 7 and 10, k_p and γ are 0.3628 and 0.0021 respectively. k_p and γ assume the values 2.5 and 0.1 respectively whenever the reference pH levels are between $6-7$ and $10-11.5$. The learning rate for the NLMS algorithm and the sampling time are set to be unity and 1 seconds respectively.

Figure 5.12 shows the performance of the adaptive neurofuzzy control scheme when the reference pH is varied periodically between 6 and 11.5. By comparing the plots in Figures 5.8, 5.10 and 5.12, it may be concluded that the initial performance

of the adaptive neurofuzzy controller pales in comparison with the Wiener-model control schemes. A plausible explanation is that a longer time is needed to learn the non-linear pH dynamics, which is more complex. With time, reasonably good tracking control is obtained for pH values between 7 and 9. The ability of the adaptive neurofuzzy controller to reject disturbances in the form of an unknown buffer is shown in Figures 5.13. Unlike the Wiener-model controllers, the adaptive neurofuzzy controller is able to prevent the carbonic acid from adversely affecting the control performance. This characteristics may be the result of the fact that the adaptive neurofuzzy controller is not constrained by erroneous *a prior* information.

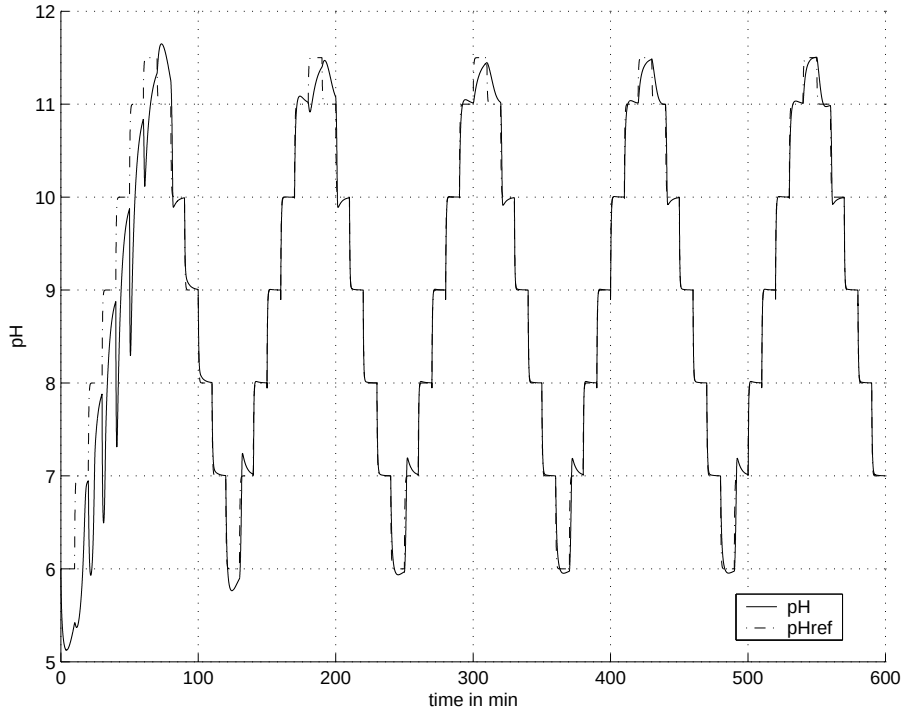

Figure 5.12. Performance of the adaptive neurofuzzy controller under nominal conditions

5.3.5 Discussions

In order to compare the performances of the three controllers objectively, the Integral Absolute Error (IAE) for successive training cycles that comprises of unit step

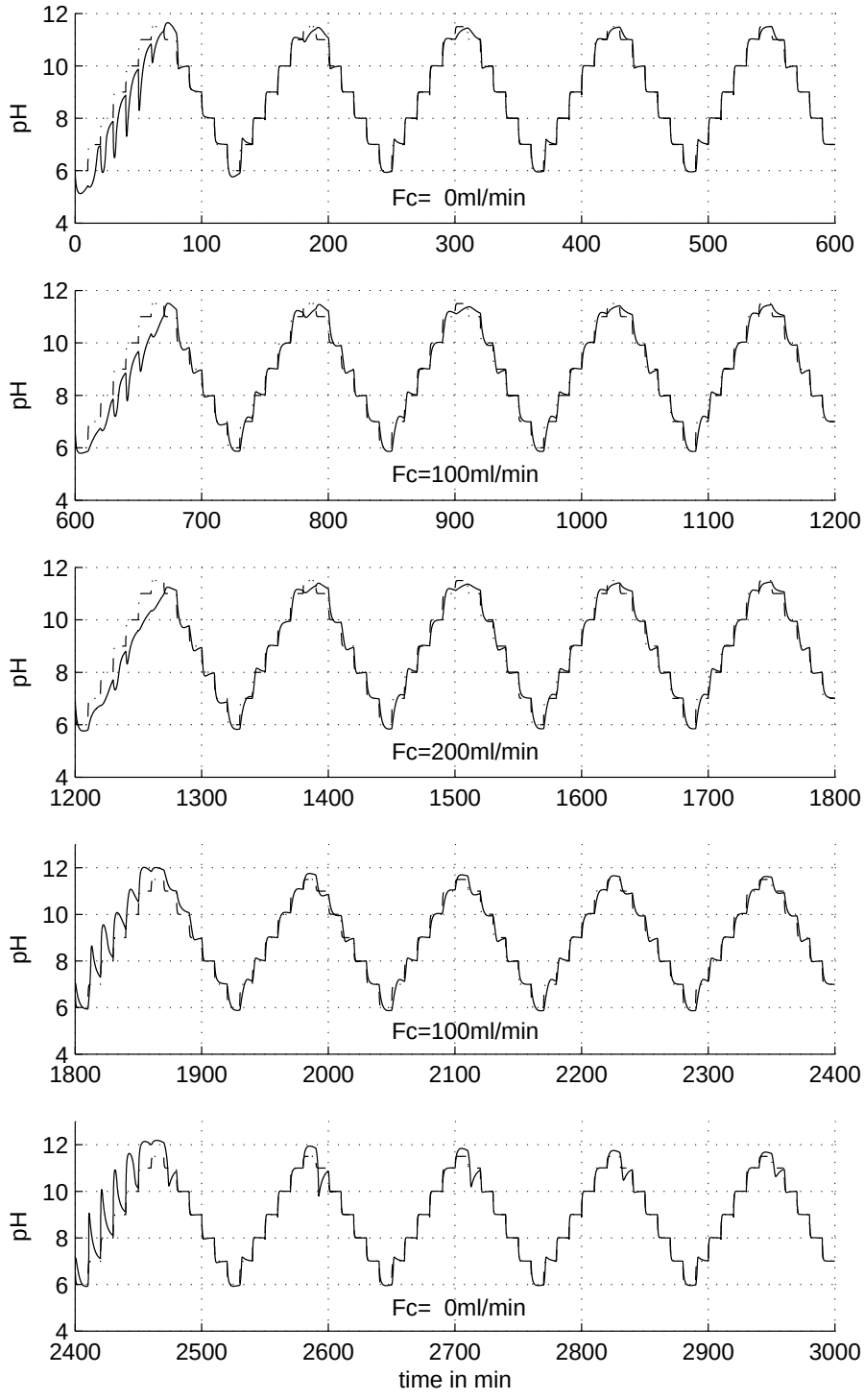

Figure 5.13. Performance of the adaptive neurofuzzy controller when an unknown buffer is introduced

changes from pH = 6 to 12 and back are shown in Figures 5.14 and 5.15. The IAE plots indicates that the "black box" approach is better during the first training cycle even though the step responses in Figures 5.8 and 5.10 appears to be better than the one in Figure 5.12. One reason behind the poorer performance of the adaptive Wiener model controller may be that the neurofuzzy controller is used to regulate base ionic concentration when it is trained using a feedback error learning rule that is based on the difference between the desired and actual pH level. When the three controllers have "learnt" the plant dynamics, the "black box" approach still provided the best performance. The same conclusions can also be drawn from the simulations obtained when the characteristics of the pH process were altered by an unknown buffer. Thus, the study seems to suggest that the addition of inaccurate *a priori* information into control structures may hinder the ability of the controller to adapt to process variations. However, generic information, such as the regions where process is sensitive/insensitive to the input, may be used to fine-tune the controller parameters.

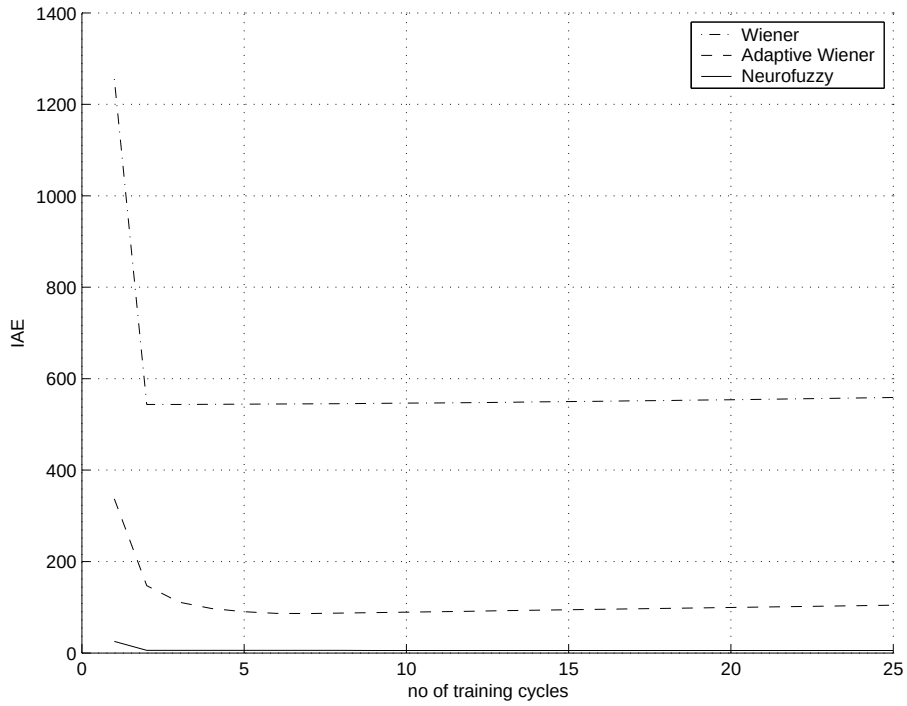

Figure 5.14. Comparison of IAE between the three controllers under nominal conditions

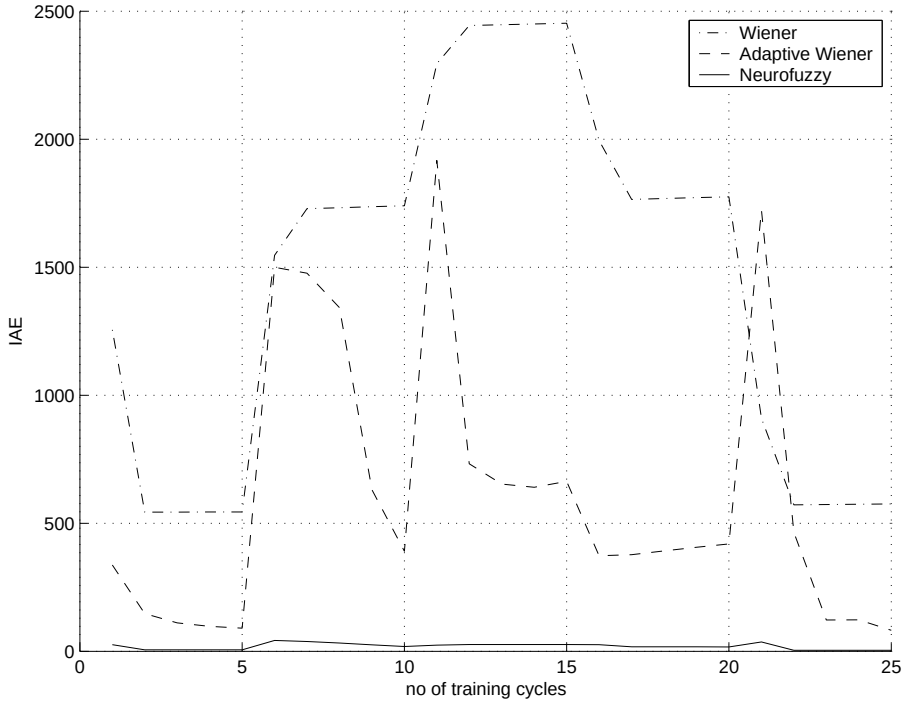

Figure 5.15. Comparison of IAE between the three controllers in the presence of unknown buffers

5.4 Experiments on the pilot pH plant

The simulation results show that the 'black box' neurofuzzy control scheme has the capability to successfully control the theoretical pH plant model. In this section, results of experiments carried out to verify the feasibility of this control scheme are detailed.

5.4.1 The pilot pH plant

Plant schematic

The pilot pH plant that is used to perform the experiments is a custom built plant (Looi, 1995). The schematic is shown in Figure 5.16.

Storage Tanks *T1* and *T2* store the process reagents CH_3COOH (acid) and $NaOH$ (base) respectively. The reagents are pumped into the mixing tank *T3*, where the pH process takes place. A stirrer is used to achieve fast mixing in order

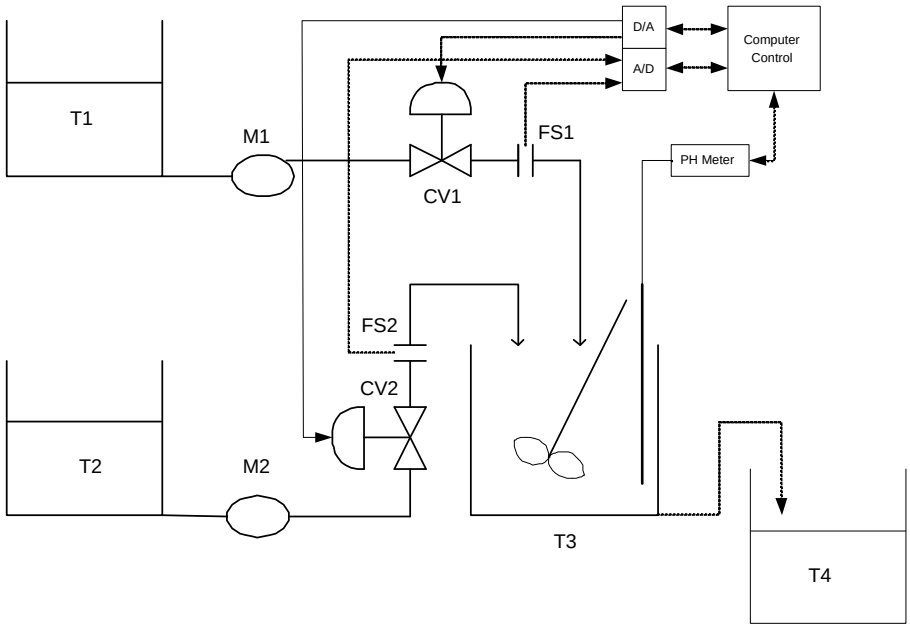

Figure 5.16. The pilot pH plant CSTR configuration

to satisfy one of the assumptions used to derive the CSTR model. A hole near the bottom of the mixing tank leads the effluent to the storage tank *T4* to keep the volume of the mixing tank constant.

Control of the rate at which the reagents flow into the mixing tank is achieved by manipulating the control valves $CV1$ and $CV2$ installed along the reagents' flow paths from the storage tank to the mixing tank. Low flow sensors ($FS1$ and $FS2$) are also installed along each flow path to measure the reagent flow rate. The outputs of the flow sensors are linked to a AD/DA card connected to a computer that acts as the digital controller to control the flow rates of the reagents via the control valves. Next, a short description on the components used to implement this schematic is detailed.

Plant components

Pump

The pump used for both the acid and base flow stream is the *Nikkiso Eiko MAGPON CP 10* magnetic drive centrifugal pump. With a rating of $6.5 liters/min$

at a head measurement of $1.5m$, the pump is oversized for the experimental setup, as the required flow rate for both the acid and the base is at about 340 and 200 ml/min.

Control Valve

The control valve used to regulate the acid flow is the *Keystone F382* ball valve ($C_v = 8$) and the $F777$ electrical actuator. It is oversized for dispensing the required amount of acid, and is not suitable for accurate dosing of the reagent. However, this is not a major problem, as the acid flow rate is kept constant during the experiments.

The base stream, which is the titrating reagent used to control the pH, requires a much more accurate positioning of the control valve to achieve minimal hysteresis error. Here, the $JordanMV1005$ linear control valve that has a C_v rating of 0.1 is used. The actuator accepts an input current between 4-20mA as the command signal.

Flow Sensor

The *Omega FLR1010* low flow sensor is used to measure both the acid and base flow rates. It has a linearity and repeatability of $\pm 3\%$ and $\pm 0.2\%$ full scale respectively, and has a temperature sensitivity of $\pm 0.2^oC$ for flow rates between 60 to 1000ml/min. It provides an output signal of 0-5V, and is powered by a 12V DC source.

pH sensor

The pH sensor used is *Orion model 290A* pH meter. It is a hand-held meter with a resolution of 0.001 pH unit and a relative accuracy of ± 0.005 pH unit. It sends the pH readings to the computer, which acts as the digital controller, directly via serial communication through its built-in RS232 port. A null modem cable is used to connect the two devices.

Stirrer

Process agitation is provided by the *Cole Parmer STIR-pAK* laboratory stirrer. It consists of a adjustable speed motor and a 2 inch three blade propeller. The motor is rated at 5000 rpm.

Digital Controller

The outputs of the flow sensors are linked to a *National Instruments PCI-MIO-16E* AD/DA card connected to a *Intel Pentium 4* computer that acts as the digital controller to control the flow rates of the reagents through command signals to the control valves. As the AD/DA card can only can output voltage signals, a *Asahi Keiki TZ-56* magnetic transducer is used to map the 0-5V output from the AD/DA card to the required 4-20mA current.

Characterization of Plant

Having described in detail the plant and its various components, the actual working performance of the plant is now described. This is an important step to take before actual control takes place, as it provides information about the plant's characteristics. Furthermore, the noise and disturbances experienced by the physical plant can be identified, and compensated for to minimize their effects on the performance of the overall system.

First, an attempt is made to determine the amount of hysteresis in the valves used by gradually increasing the voltage applied to the control valve in steps of 0.1V and observing the changes to the output from the flow sensor, which has a linearity of 0.25% full scale. Figure 5.17 gives an idea of the relationship between the flow rate and the control voltage applied to the acid control valve. The deadband for the acid valve is very large, with about 3.1V needed before any flow can occur. Also, changes to the flow rates occur only when a minimum of 0.2V difference in the control output voltage is applied. Coupled with the fact that the maximum voltage that can be applied to the pump is 4.5V due to the limitation on the flow sensor's measurement range, the working range for the control valve is extremely limited. It is seen that the hysteresis for the acid flow control valve, which is derived by taking the maximum difference between the measured flow rates between the opening and closing paths, over the input range is about 82%. Another observation is that the operation of the valve is not repeatable even when the initial conditions are the same, so the same change in output control voltage can lead to vastly different

flow rates. Fortunately, the acid flow rate is not the control signal. As F_a is fixed at 340 ml/min for the duration of the experiment, open loop manual trial and error tuning of the control valve was performed at the start of each experiment to obtain the required flow rate. The main reasons for the poor performance of this valve arises from its poor construction, the fact that it is grossly oversized for the required operation, as well as poor maintenance of the valve through the years.

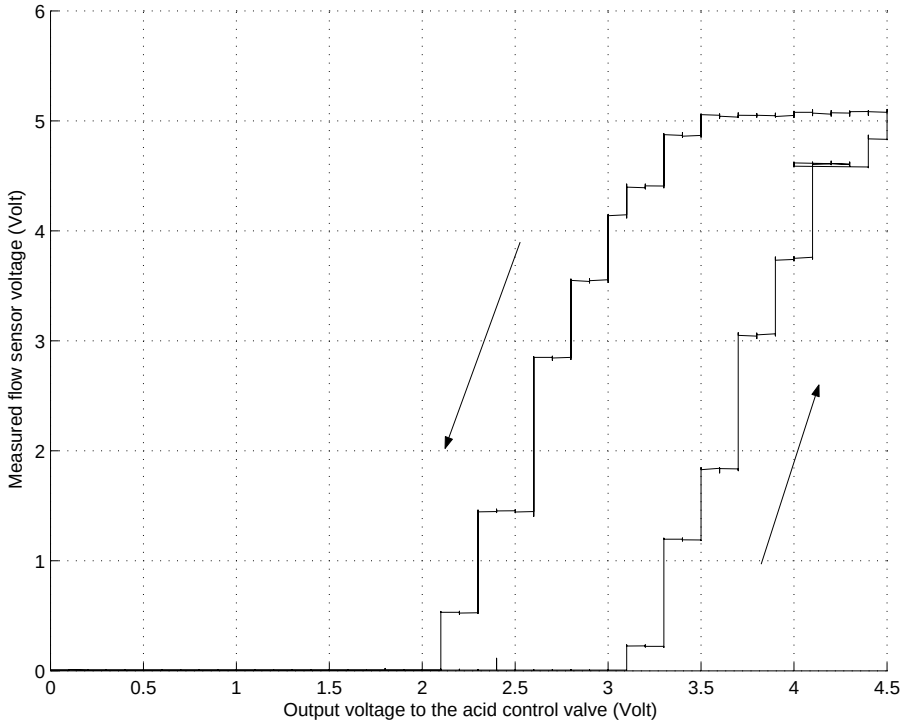

Figure 5.17. Hysteresis plot for the acid control valve

Compared to the acid control valve, the base control valve has better characteristics (see Figure 5.18). With the deadband estimated to be 1.8V, the working range is significantly larger than the acid control valve. Most importantly, the hysteresis is about 4.8%. Though it still exceeds the recommendation of less than 1% hysteresis to achieve good control (Buckbee, 2001), reasonable performance can be obtained by placing the valve under a PI controller. The gain and integral time of the PI controller is 0.007 and 3 seconds respectively.

Next, a noise analysis is performed on the base control flow sensor's readings using FFT, and the magnitude plot is displayed in Figure 5.19. It can be seen that

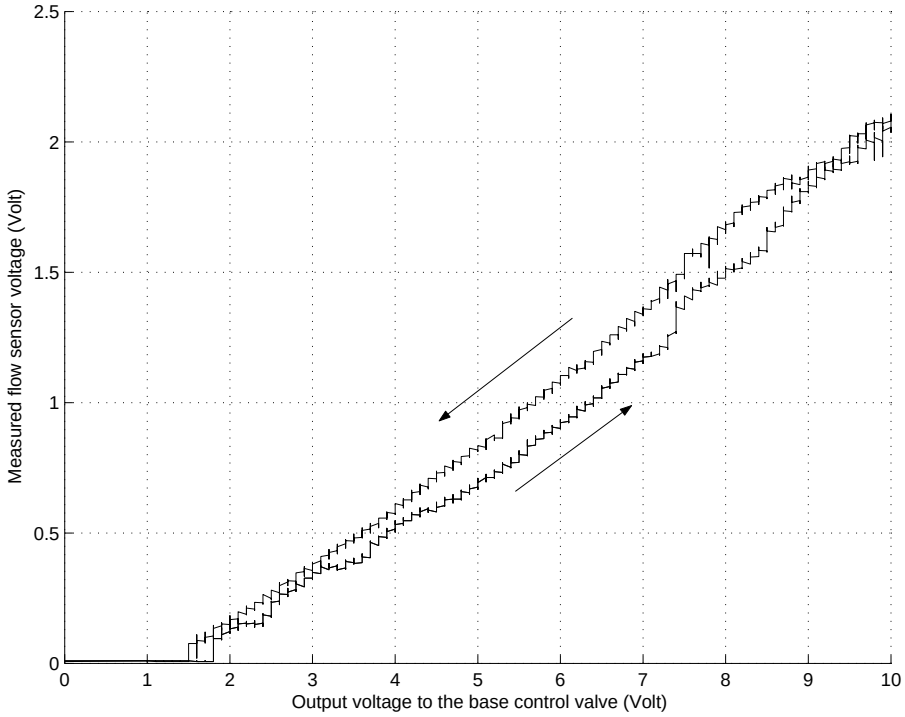

Figure 5.18. Hysteresis plot for the base control valve

the noise frequencies occur at about 100Hz, 150Hz, and 300Hz. To minimize the effects of noise on the control system, over-sampling is first performed at 50Hz. The over-sampled data are then sorted, and the middle 60% of the samples are then averaged to obtain the measured flow rate after conversion. This choice of taking the average is to reduce the variance of the measured flow rate, while taking the median removes the erroneous low frequency 'spikes' from the measured data. A exponentially moving average filter with a window length of 2 and filter constant of 0.7 is then applied to the measured data to further reduce the noise in the flow rate reading.

The last component of the test rig is the pH sensor. Since the pH meter used is able communicate digitally with the computer through the serial communication, there is no concern over the possibility of interference with conversions between analogue and digital formats. Although the sensor is accurate (from manufacturer's specifications), the dynamics of the sensor is too slow for control purposes as it is more suited to measure stationary pH valves than rapidly changing ones. Typically,

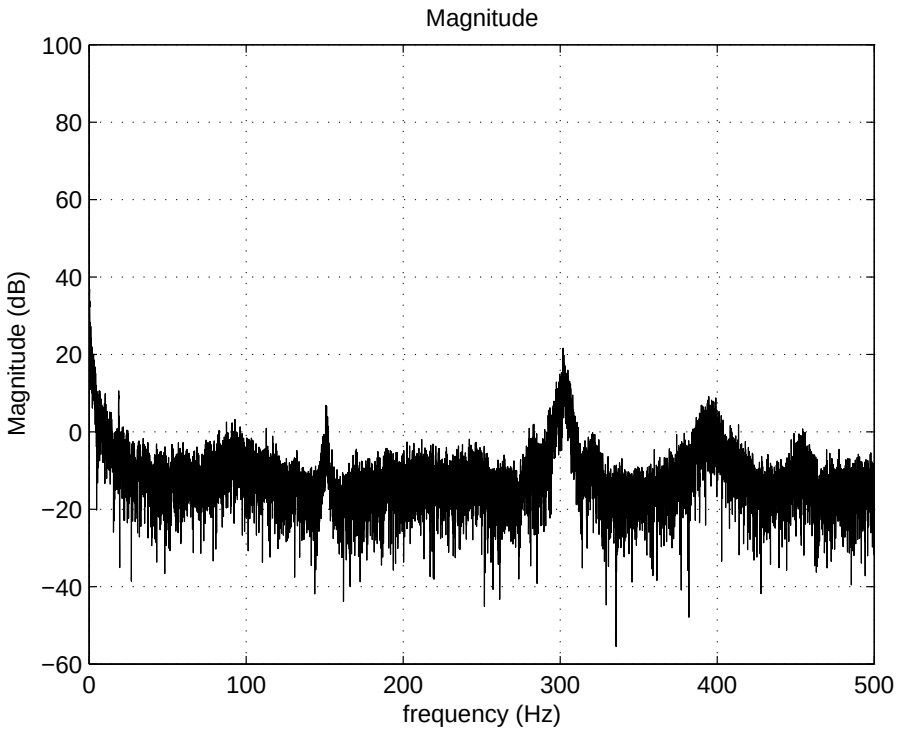

Figure 5.19. FFT Magnitude plot on the base flow sensor input

more than 3 minutes passes before the pH sensor reading stabilizes. Maybe due to this reason, the device data sheet states that the fastest rate at which the sensor can transmit its readings to the computer is 5 seconds. This is too slow for neurofuzzy control scheme. By tweaking the communication protocol, a sampling rate of 2.5 seconds is achieved. Another problem is that the pH sensor fails intermittently, and a pH value of 0 is returned. A crude way that is used to alleviate this is to use the previous measured pH value to do control. To reduce the measurement noise, a moving average filter of length 10 is used to filter the pH measurements.

5.4.2 Experiment

Due to the limitations of the hardware, a number of specifications that are used in the simulations are not achievable. Of particular importance is the inability to sample at the required frequency. Simulations reveal that a sampling time of 1 second is needed to prevent oscillatory output pH responses, yet in the actual

experiments, only a sampling period of 2.5 seconds is possible. In a bid to accommodate the slower sampling rate, the reference model's time constant is slowed to 60 seconds to allow for smaller changes in the reference pH. The acid flow rate is also changed from 400 ml/min to approximately 340 ml/min due to the difficulty in obtaining the flow rate used in the simulation study. The proportional gain kp, is changed to 0.1 and γ to 0.004 to reflect the changes.

Figure 5.20 shows the simulated response with the experimental setup. A very small measurement noise with a power spectral density of 0.01 was added to the simulation pH output. Basically, an overshoot is seen at pH 7 and 10, and slight oscillatory response seen at pH levels 8 and 9 due to the sensitivity in these regions. Overall, reasonable control is obtained.

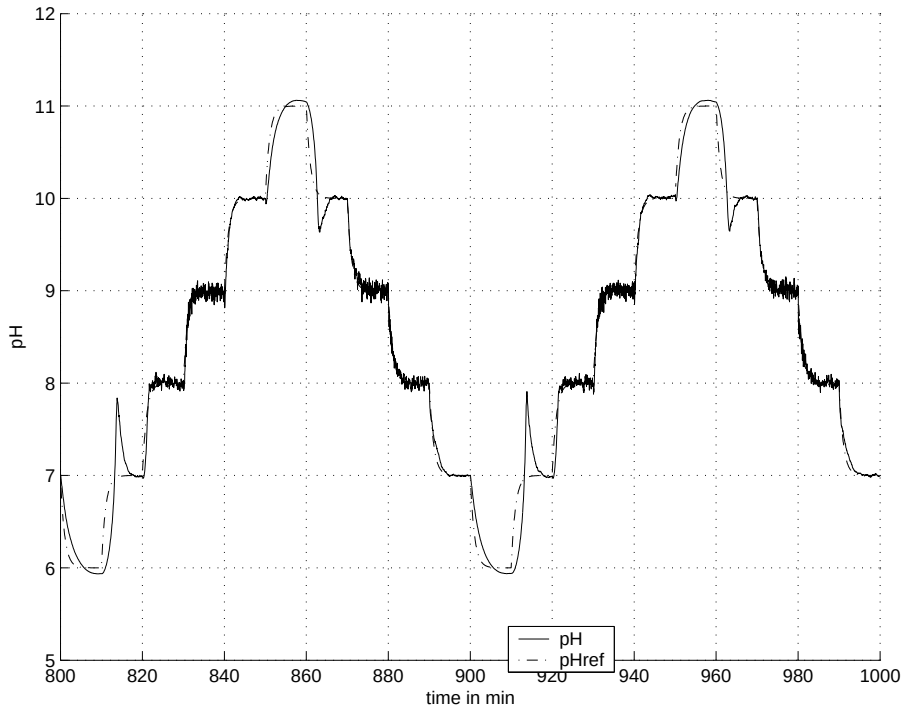

Figure 5.20. Simulation results using the experiment controller's parameters

In order to speed up the learning process, the weights obtained using the simulated pH plant are used to initialize the neurofuzzy controller's weights. Figure 5.21 displays the experimental control response that is obtained. It resembles the simulated response. As expected from the simulation results (Figure 5.20), a slight

overshoot is obtained at pH 7, and mild oscillations observed at pH 8. The rise from pH 10 to pH 11 is slow due to a marginally slower response in the actual base reagent's flow rate compared to the desired output flow rate. Despite applying many noise reduction techniques on the flow rate measurements and much work in trying to obtain good control over the base control valve, the measured flow rate still exhibits "spiky" behaviour due to the inability to get the control valve to achieve the required preciseness needed to control the pH level in the sensitive regions (see Figure 5.22). This problem is also the reason why the output response around pH 8 is oscillatory. Overall, the neurofuzzy control scheme is shown to be able to provide reasonable control of the actual pH plant.

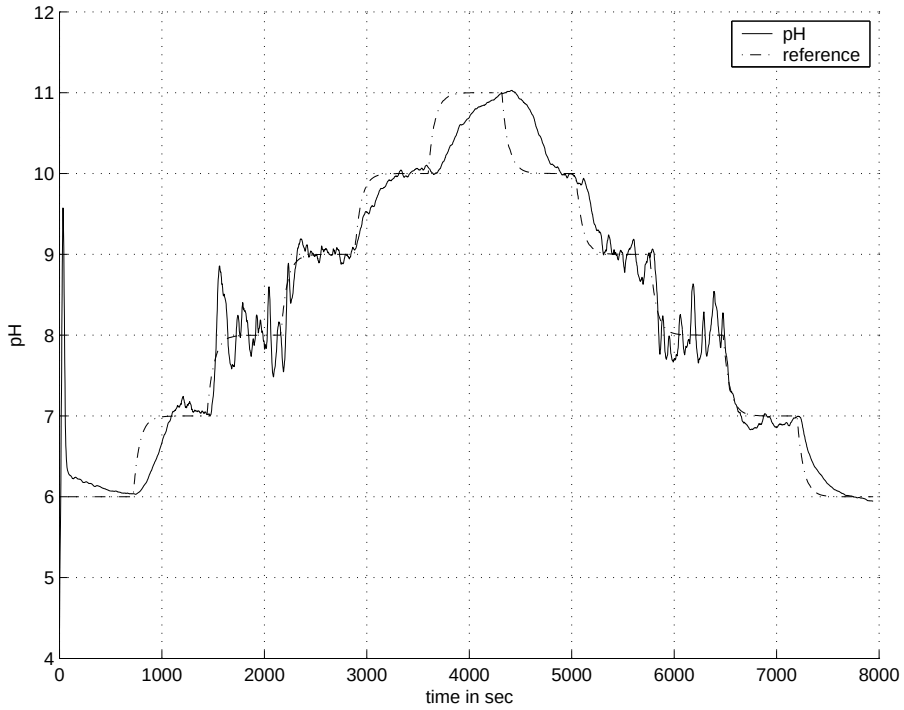

Figure 5.21. Control performance in the pH experiment

5.5 Conclusion

Control of a pH process using a neurofuzzy controller has been simulated. Comparisons made show that using *a prior* structural knowledge in the form of an

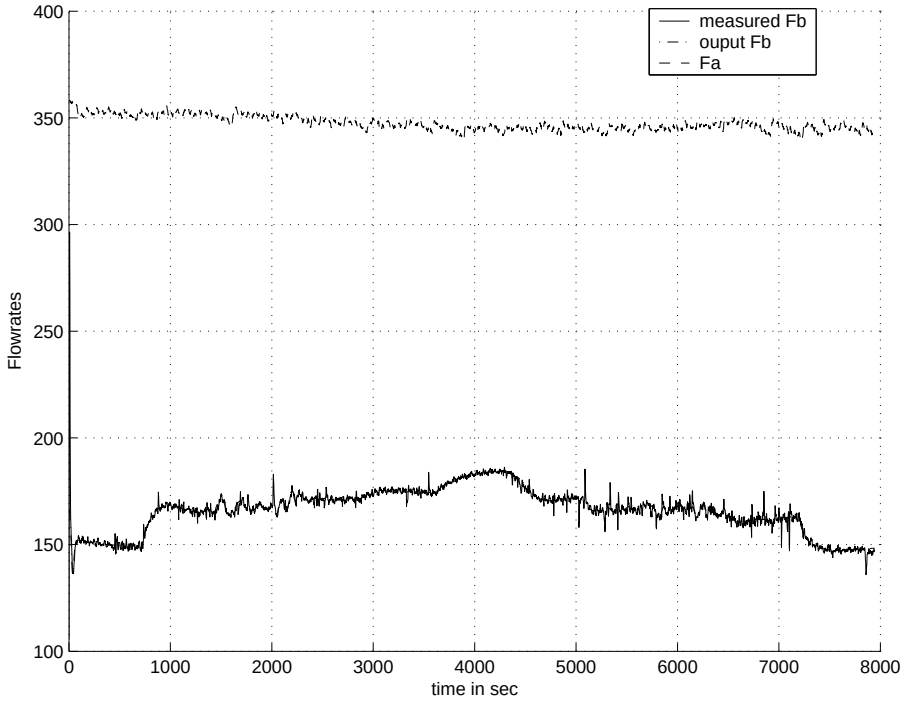

Figure 5.22. Flow rates in the pH experiment

inverse model leads to poorer control performance under varying buffering conditions. Experiments are also carried out to test the feasibility of the neurofuzzy control scheme on a pilot pH plant.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Much work has been performed on developing the self-learning neurofuzzy control scheme in this thesis. First, a stability guide for the neurofuzzy control scheme is established from insights gained by examining the stability of the learning algorithms individually. Simulations results verified the feasibility of the stability criteria.

A comparison of the various feedback error learning strategies is performed using a liquid level plant as a test bed. An attempt is made to compare the performance of an alternative commissioning strategy's performance for the proposed strategy with the original one. The study show that the alternative method offers performance comparable with the original one. Simulation results also show that the proposed FELS's performance is superior to the other learning strategies, while experimental results demonstrate its feasibility in real world conditions.

As much as the incorporation of *a prior* information about the process may bring about more "intelligent" controllers, there is the associated difficulty in ascertaining the information's accuracy when the process dynamics changes drastically. The pH neutralization process, with its severe nonlinearity and sensitivity, is used to test whether there is merit in including structural information into the control scheme. While the control task is simplified by the inclusion, difficulties with cop-

ing with changes to the buffering conditions makes the inclusion undesirable. Even with adaption of the structural information on-line, simulation results show that the neurofuzzy control scheme is able to cope best without using the structural information. The feasibility of the neurofuzzy control scheme in handling an actual pH process is also verified experimentally.

6.2 Suggestions for Future Work

Much is still needed to improve on the work that has been reported here. First, a study to improve of the effects of the two intertwined optimization algorithms in the learning mechanism on each other may bring about a better and more intuitive understanding of the neurofuzzy control scheme to improve the rate at which the weights converges.

Presently, the reference model in the control scheme presents an extra degree of freedom to tune the control scheme for performance. However, at the same time, it is used to prevent the unrealizable set point changes from corrupting the neurofuzzy controller's weights, resulting in the loss of that freedom. Work may be expanded to see if there are other ways to incorporate the constraints into the control scheme to regain that degree of freedom, which may then be used to improve control performance.

There is also a need to extend the commissioning guide to deal with plants that may not be well controlled by PI/ PID controllers. When dealing with highly nonlinear plants, the choice of tuning parameters for many control schemes is left to trial and error. Though the effects of each tuning parameter for the neurofuzzy control scheme has been understood and documented for linear plants, the effect of nonlinearities on each tuning parameter, and the stabilization of the control scheme is still open for discovery. The extension of the stability analysis to nonlinear systems can also be investigated.

Bibliography

Abonyi, J., L. Nagy and L. Szeifert (2001). Fuzzy model-based predictive control by instantaneous linearization. *Fuzzy Sets and Systems* **120**, 109–122.

Astrom, K.J. and B. Wittenmark (1995). *Adaptive Control*. 2nd ed. ed.. Addison Wesley.

Brandizzi, J., M. Santos and A.L. Dexter (1999). Commissioning an adaptive fuzzy control scheme. In: *Proceedings of the 6th UK workshop on Fuzzy Systems*. London,UK. pp. 125–132.

Brown, M. and C. Harris (1994). *Neurofuzzy Adaptive Modelling and Control*. Prentice Hall. UK.

Bruske, J., M. Hansen, L. Riehn and G. Sommer (1997). Biologically inspired calibration-free adaptive saccade control of a binocular camera-head. *Biological Cybernetics* **77**, 433–446.

Buckbee, G (2001). How to identify and troubleshoot control valve problems 'on the fly'. *Control Engineering*.

Clarke, D.W. (1984). Pid algorithms and their computer implementation. *Transactions of the Institution of Measurement and Control* **6**(6), 308–316.

Cox, M.G. (1972). The numerical evaluation of b-splines. *J. Inst. Math. Appl.* **10**, 134–149.

Czogala, E. and W. Pedrycz (1981). On identification in fuzzy systems and its applications in control problems. *Fuzzy Sets and Systems* **6**, 73–83.

Edgar, C.R. and B.E. Postlethwaite (2000). Mimo fuzzy internal model control. *Automatica* **34**, 867–877.

Gomi, H. and M. Kawato (1993). Neural network control for a closed-loop system using feedback error learning. *Neural Networks* **6**(7), 933–846.

Haykin, S. (1999). *Neural networks : a comprehensive foundation*. 2nd ed.. Prentice Hall. Upper Saddle River.

Jacobs, R.A. and M.I. Jordan (1993). Learning piecewise control strategies in a modular neural network for control systems. *IEEE Transactions on Systems, Man and Cybernetics* **23**(2), 337–345.

Kawato, M., K. Furukawa and R. Suzuki (1987). A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernatics* **57**, 169–185.

Kim, S. W., and J. J. Lee (1996). Filtered-error-learning neural networks for stable trajectory tracking control of robot manipulators. *Mechatronics* **6**(2), 181–192.

Kraft, L. G. and D. P. Campagna (1990). A comparison between cmac neural network control and two traditional adaptive control systems. *IEEE Control Systems Magazine* pp. 36–43.

Linkens, D.A. and S. Kandiah (1996). Long-range predictive control using fuzzy process models. *Chem Eng Res Des* **74**, 77–88.

Lo, C.H. (2001). Development of the feedback error learning strategy for training neurofuzzy controllers. B.Eng thesis, National University of Singapore.

Lo, C.H. and W.W. Tan (2001*b*). Development of feedback error learning strategies for training neurofuzzy controllers on-line. In: *CIRAS*. pp. 94–99.

Looi, K.O. (1995). Neural network strategies in modelling and control of a ph process. Master's thesis. National University of Singapore.

McAvoy, T.J., E. Hsu and S. Lowenthal (1972). Dynamics of ph in controlled stirred tank reactor. *Industrial Engineering Chemistry Process Design and Development* **11**(1), 68–70.

Ohno, H., K. Suzuki and A. and Aoki (1994). Neural network control for automatic control system. *Neural Networks* **7**(8), 1303–1312.

Postlethwaite, B.E. (1993). A model-based fuzzy controller. *Chem Eng Res Des* **72**, 38–46.

Postlethwaite, B.E., M. Brown and C.H. Sing (1997). A new identification algorithm for fuzzy relationional models and its application to model-based control. *Chem Eng Res Des* **75**, 453–458.

Santos, M. and A.L. Dexter (2001). Temperature control in a helium cryostat using a self-learning neurofuzzy controller. *IEE Proceedings on Control Theory and Applications* **148**(3), 233–238.

Santos, M., J. Brandizzi and A.L. Dexter (2000). Control of a cryogenic process using a fuzzy pid scheduler. In: *Proceedings of the IFAC Workshop on Digital Control : Past, Present and Future of PID Control*. Terrassa, Spain. pp. 401–405.

Seborg, D.E., T.F. Edgar and D.A. Mellichamp (1989). *Process Dynamics and Control*. John Wiley. USA.

Stephanopoulos, G (1984). *Chemical Process Control: an introductin to theory and practice*. Prentice Hall. Englewood Cliffs, N.J.

Tan, W.W. (1997). Self-learning neurofuzzy control of non-linear systems. PhD thesis. University of Oxford.

Tan, W.W. and A.L. Dexter (1999). Self-learning neurofuzzy control of a liquid helium cryostat. *Control Eng. Practice* **7**(10), 1209–1220.

Tan, W.W. and C.H. Lo (2001*a*). Development of feedback error learning strategies for training neurofuzzy controllers on-line. In: *IEEE 10th International Fuzzy Systems Conference*. Vol. 2. pp. 1016–1021.

Tzafestas, S.G., G.G. Rigatos and E.J. Kyriannakis (1997). Geometry and thermal regulation of gma welding via conventional and neural adaptive control. *Journal of Intelligent and Robotic Systems* **19**, 153–186.

Velthuis, W.J.R and de Vrie (2000). Stability analysis of learning feed-forward control. *Automatica* **36**, 1889–1895.

Wang, L.X. (1992). Fuzzy systems are universal approximators. In: *IEEE International Conference on Fuzzy Systems*. pp. 1163 –1170.

Wang, L.X. (1994). *Adaptive Fuzzy Systems and Control: Design and Analysis*. Prentice Hall. USA.

Wang, L.X. (1997). *A course in fuzzy systems and control*. Prentice Hall. USA.

Widrow, B. and E. Walach (1996). *Adaptive Inverse Control*. Prentice Hall. USA.

Wright, R.A (1998). On-line identification and nonlinear control of ph process. *Ind. Eng. Che. Res.* **37**, 2446–2461.

Author's Publications

List of publications

[1] Tan, W.W. and Lo, C.H. (2001*a*). Development of Feedback Error Learning Strategies for Training Neurofuzzy Controllers On-line. *IEEE 10th International Fuzzy Systems Conference*, 2001, Vol. 2, 1016-1021.

[2] Tan, W.W. and Lo, C.H. (2001*b*). On Utilising Structural Information for Adaptive Control of a pH Neutralisation Process. *CIRAS*, 94-99

[3] Tan, W.W. and Lo, C.H. (2003). Development of Feedback Error Learning Strategies for Training Neurofuzzy Controllers On-line. *to be presented at European Control Conference, Cambridge, U.K. September 2003*