2004-11-01

Local Model Network Application in Control

Ruiyao Gao
Technological University Dublin

Local Model Network

Applications in Control

Doctor of Philosophy (Ph.D)

School of Control Systems and Electrical Engineering

Dublin Institute of Technology

Ruiyao Gao M.Eng, B.Eng.

Supervisors: Dr. Aidan O'Dwyer,

Dr. Eugene Coyle, Dr. Séamus McLoone

November 2004.

ABSTRACT

The local model (LM) network is considered for the control of complex nonlinear systems. Both controller design and system analysis techniques are investigated for the purpose of the development of an overall global controller with guaranteed stability and performance, based on the control methods and theories well developed for linear systems. In particular, the influence of the offset term of affine LM networks on the performance and stability of closed-loop systems is investigated.

Assuming the system changes 'slowly' enough, an integrator can be utilised in the controller design by considering the offset term as 'constant'. Gain-scheduled local controller (LC) networks based on feedback control and generalised predictive control methods are proposed for the control of a highly nonlinear simulated process, the continuous stirred tank reactor. Test results show reasonably good performance, but also expose the weakness of the interpolation procedure.

Considering the nonlinear dynamics inherent in the interpolation procedure, the stability issues associated with blending affine systems are investigated. Assuming the corresponding linear blending system is exponentially stable, the affine blending system will be bounded. The ultimate bound is determined for both open loop systems and closed-loop systems via affine state feedback control.

The velocity-based (VB) LM network has enhanced the capability to capture the dynamics of nonlinear systems compared to normal LM networks. To have the best access to dynamical information from the VB model, a state feedback integral controller is skilfully proposed for the controller design. This approach overcomes the difficulty in the practical implementation of velocity-based approaches. In addition, a discrete-time version of the velocity-based LM network structure is proposed for the purpose of extending the VB approach to the discrete time domain.

Declaration

I certify that this thesis which I now submit for examination for the award of Ph.D. Degree, is entirely my own work and has not been taken from the work of others save and to extent that such work has been cited and acknowledged within the text of my work.

This thesis was prepared according to the regulations for postgraduate studies by research of the Dublin Institute of Technology and has not been submitted in whole or in part for an award in any other Institute or University.

The work reported on in this thesis conforms to the principles and requirements of the Institute's guidelines for ethics in research.

The Institute has permission to keep, to lend or to copy this thesis in whole or in part, on condition that any such use of the material of the thesis be duly acknowledged.

Signature ______________ Date Nov. 14th. 2004

Candidate

Acknowledgement

Firstly, I wish to express my gratitude to my supervisor Dr. Aidan O'Dwyer for his support and help throughout this thesis. His advice and guidance was invaluable on many occasions. Also I wish to thank the co-supervisors Dr. Séamus McLoone and Dr. Eugene Coyle for their cordial help and support.

I would also like to thank all the staff and Postgraduates of the School of Control Systems and Electrical Engineering who have helped me on the way. Special thanks go to, Michael, Richard, Finbarr, Eileen, Tony, Mark, Pauline and Steven for both technical and spiritual advice all throughout the years.

Furthermore, I want to acknowledge the support of my friends both inside and outside the department Joe, Andreas, James, Kennenth and Elmar. To all of my friends, thank you for your help and friendship.

Additionally I would like to thank the Staff and Postgraduates in the Intelligent Systems and Control Research Group in Queens University Belfast, who made my weeklong stay enjoyable and productive.

Finally, I wish to thank my parents for the support they have given me over all the years.

Furthermore, I would like to acknowledge the funding received through the Dublin Institute of Technology. Special thanks go to Dr. Jonathan Fisher for his full support in research funding.

Nomenclature

All symbols and abbreviations in this thesis are defined in the text as they occur. A summary of the main symbols and abbreviations is provided here for ease of reference.

Main Symbols

Neural networks

x	Input vector
y	Output vector
u	Control input
w_i	i^{th} layer weight vector
b_i	i^{th} layer threshold/bias value
ϕ_i	i^{th} layer nonlinear operator (sigmoid function)
N_i	Number of neuron inputs
N_h	Number of hidden layer neuron
d	Desired system output
σ_i	Width factor in radial basis function (RBF) network

Generalised predictive control (GPC):

H_p	Maximum prediction horizon
H_m	Minimum prediction horizon
H_C	Control horizon

λ	Control weighting sequence
$y(k)$	System output
$w(k)$	Desired output/ Reference set point
Δ	The difference operator (discrete time domain)

Local model (LM), local control (LC) networks and velocity-based multiple model networks:

Z	Operating space
N	Number of local models
$\hat{f}_i$	Local model function
$\psi(t)$	Scheduling vector
c	Centre of the basis function
σ	Width of the basis function
ρ	Normalized validity/weighting function
v	Un-normalised weighting function
$\phi^C(t)$	Controller information vector
δu	Perturbation system input
$\phi(k-1)$	Discrete time domain information vector for input/output model
$\phi_{ar}^T(k-1)$	Information vector for input/output ARX model
θ	Parameter vector
$h(o)$	High order modelling error
α	Offset term in ALM
x	State of general nonlinear system
$\tilde{x}$	State of multiple model system
$\bar{x}$	State of local models
$\bar{w}$	Differential of the state of local models
w	Differential of the state of general nonlinear system
r	Reference signal
y	System output
u	Control input
K	State feed back gain

Bold characters denote vectors and matrices; subscription 'i' denotes the i^{th} local model or local controller; subscription 'o' denotes the operating points where the linearization is formulated; '^' denotes the predicted signal; '(t)' denotes signals in the continuous time domain and '(k)' denotes signals in the discrete time domain.

Notes: There are many approaches have been concerned both in modelling and control of the nonlinear processes. It might happen that some symbols mean differently at different places. This is to keep the consistency with the common symbols that those approaches normally employ. For those symbols, please refer to the context.

Main Abbreviations:

ARX	Autoregressive with exogeneous inputs
CSTR	Continuous Stirred Tank Reactor
CARIMA	Controlled Auto-Regressive and Integrated Moving Average model
LC	Local Control (network)
LLM	Local Linear Model (network)
ALM	Affine local model (network)
MLP	Multi-layer Perceptron
NARMAX	Nonlinear AutoRegressive Moving Average with Exgeneous inputs
PID	Proportional Integral Derivative Controller
RBF	Radial Basis Function
SOM	Self-Organising Maps
TS	Takagi-Sugeno (fuzzy model)
GPC	generalized predictive control
MBPC	Model based predictive control
VB	Velocity based (multiple model networks)
LPV	linear parameter varying

Table of Contents

List of Figures

Chapter 1

Chapter 2

Chapter 3

Chapter 5

List of Figures

Chapter 1

Chapter 2

Chapter 1

Introduction

Non-linear processes, by their nature, are non-uniform and invariably require custom designed control schemes to deal with their individual characteristics. No general theory deals comprehensively with the wide range of non-linear systems encountered. In an attempt to accurately model nonlinear dynamical systems, a wide variety of techniques have been developed, such as multiple layer perceptron (MLP) neural networks (Narendra and Parthasarathy, 1990), radial basis function (RBF) networks (Moody and Darken, 1989), nonlinear auto-regressive moving average with exogeneous inputs (NARMAX) models (Chen and Billings, 1989), Weiner models (Schetzen, 1981) and Hammerstein models (Billings and Fakhouri, 1982). While the accuracy of such models offer a potentially significant improvement over linear models, the process control engineer is faced with the difficulty of their more-or-less so-called black-box representation of the dynamics of nonlinear systems. These types of model fail to exploit the significant theoretical results available in the conventional linear modelling and control domain, and make it difficult to analyse the behaviour of the controlled systems, and to prove their stability. This chapter first briefly reviews the neural networks application in modelling and control, then discusses the main topic of local model networks application in modelling and control for nonlinear systems.

1.1 Neural Networks for Nonlinear Modelling

The basic concept of artificial neural networks stems from the idea of mimicking individual brain cells or neurons in a fairly simple way, and then connecting these models in parallel to offer a complex processing mechanism, which exhibits learning and training in terms of its overall nonlinear characteristics.

The earliest work in neural computing goes back to the 1940's, when McCulloch and Pitts (McCulloch and Pitts, 1943) introduced the first neural network computational model. In the 1950's, Rosenblatt's work resulted in a two-layer network, the perceptron, which was capable of learning certain classifications by adjusting connection weights (Rosenblatt, 1959). The perceptron laid the foundations for later work in neural computing. In the 1970's, Grossberg (Grossberg, 1973, 1978) developed adaptive resonance theory (ART) to deal with the problem of governing neuron biological systems. However, a real resurgence in this field began in the 1980's with the work done by Hopfield (Hopfield, 1982) and the self-organizing map developed by Kohonen (Kohonen, 1982). Shortly afterwards, Rumulhart and McClelland (Rumulhart and McClelland, 1986) rediscovered the back-propagation learning algorithm first developed by (Werbos, 1974) and brought the research in this area to a new stage. Subsequently, multi-layer perceptrons (MLPs) have very quickly become the most widely encountered artificial neural networks, particularly in the area of systems and control (Narendra and Parthasarathy, 1990, Hunt et al, 1992, Irwin et al., 1995, Agarwal, 1997). Meanwhile, a relatively simple neural network, the RBF network, attracted a lot of research attention since the beginning of the 1990's. The following two sections give a concise introduction to both MLP and RBF networks.

1.1.1 MLP Neural Networks

At the beginning of the 1990's, researchers have been very enthusiastic about the potential of neural networks, especially the multi-layer-perceptron (MLP) networks for nonlinear modelling and control. The reasons for this enthusiasm were several demonstrations of the universal approximation capability of the MLP. Giving the required number of neurons (the network structure), with a continuous and differential transfer function, one hidden layer neural network can approximate any continuous function to an arbitrary degree of accuracy (Hornik et al., 1989; Cybenko, 1989).

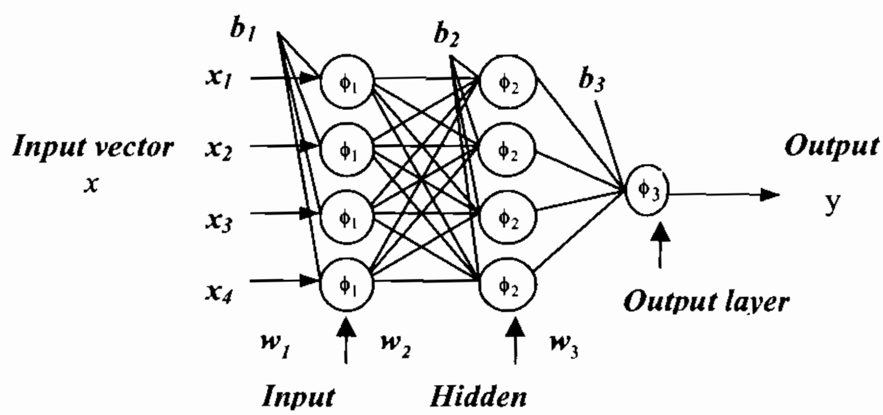


Figure 1-1. MLP neural networks

A typical feed-forward multi-layer network with an input layer, an output layer, and one hidden layer is shown in figure 1-1. Such a network can be described by means of the equation:

$$y = \phi_n(w_n\phi_{n-1}(w_{n-1}\cdots\phi_1(w_1x + b_1) + \cdots + b_{n-1}) + b_n) \qquad (\text{I}.1)$$

in which x is the input vector, y is the output vector (there is only one variable in figure 1-1), w_i is the weight vector associated with the i^{th} layer, ϕ_i is a nonlinear operator associated with the i^{th} layer and b_i indicates threshold or bias values associated with each node in the i^{th} layer. In the majority of cases, ϕ_i is a sigmoid function, such as those given in equation (1.2) or (1.3); for all layers, ϕ_i is an identical sigmoid function in input and hidden layers and a linear form in the output layer. Figure 1-2 shows an example of ϕ_i.

$$\phi(\lambda) = \frac{1}{1 + \exp(-\lambda)} \qquad (1.2)$$

$$or \qquad \phi(\lambda) = \frac{1 - \exp(-\lambda)}{1 + \exp(-\lambda)} \qquad (I.3)$$

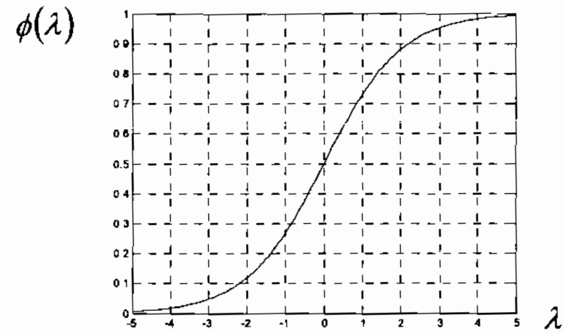


Figure 1-2. Sigmoid function

Normal use of an MLP, in practice, involves training the network on a set of desired values. During the training, the response of the output elements in the network are compared with a corresponding set of desired responses. Error signals associated with the elements of the output layer are thus readily computed, so adaptation of the output layer is straightforward. The fundamental difficulty associated with adapting a multi-layered networks lies in obtaining error signals for hidden layer neurons, that is, for neurons in layers other than the output layer. The back propagation algorithm provides a method for establishing these error signals.

Learning of the weighting parameters by a back-propagation method (Rumulhart and McClelland, 1986) is to minimize a sum square error ε^2 as defined in equation (1.4)

$$\varepsilon^2 = \frac{1}{2}\sum_{k=1}^{n}(d_k - y_k)^T (d_k - y_k) \qquad (1.4)$$

where n is the length of the training data set, d_k denotes the desired value and y_k denotes the model output.

In its simplest form, back propagation training begins by representing an input vector $\boldsymbol{x}$

to the network, sweeping forward through the system to generate an output response vector $\boldsymbol{y}$, and computing the errors at each output. Continually sweeping the effects of the errors backward through the network to associate a square error derivative δ_{ij} with neuron j in i^{th} layer, computing a gradient from each δ_{ij}, and finally updating the weights of each neuron based on the corresponding gradient. A new pattern is then presented and the process is repeated (Widrow and Lehr, 1990, Rumrlhart and McClelland, 1986).

The accuracy in modelling of complex nonlinearities is the main advantage of the use of neural networks over linear models. However, with the exception of its ability to capture the nonlinearities, the use of the MLP, in practice, implies slowness and uncertainty of learning convergence due to the steepest gradient learning method commonly used. Meanwhile, MLP neural networks have a lack of transparency in describing the system properties due to their black-box representation. These are important restrictions for the usefulness of the MLP approach in controller design, although there is some reported work to improve the training method for the purpose of overcoming the described problems.

1.1.2 RBF Networks

The radial basis function (RBF) network is another very popular network-structured modelling approach. Figure 1-3 shows the schematic diagram of an RBF network with five receptive field units.

The output of the i^{th} receptive field unit (or hidden unit) is

$$\omega_i = R_i(\boldsymbol{x}) = R_i\left(\left\|\boldsymbol{x} - \boldsymbol{c}_i\right\|^2 / \sigma_i^2\right) \qquad (1.5)$$

where $\boldsymbol{x}$ is an N-dimensional input vector, $\mathbf{c}_i$ is a vector with the same dimension as $\boldsymbol{x}$ and $R_i(\bullet)$ is the i^{th} receptive field response with a single maximum at the origin.

Typically, $R_i(\bullet)$ is chosen as a Gaussian function, which will be introduced in detail in section 2.3.3. Thus the radial basis function ω_i computed by the i^{th}-hidden units is the maximum when the input vector $\boldsymbol{x}$ is near the centre $\mathbf{c}_i$ of that unit, and is close to zero when the input vector $\boldsymbol{x}$ is far away from the centre $\mathbf{c}_i$ of that unit.

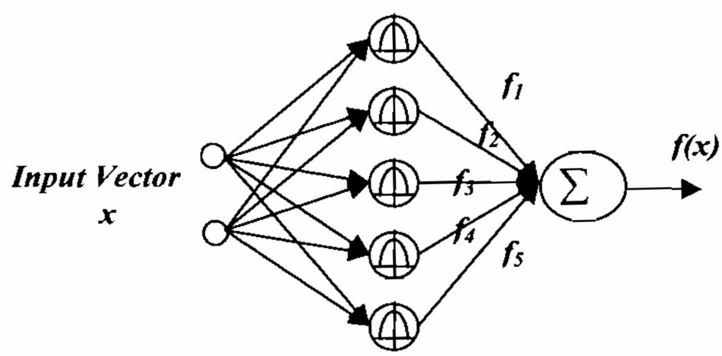


Figure 1-3. A radial basis function network

The output of an RBF network can be computed in a simpler way, as shown in Figure 1-3. It is the weighted sum of the function value associated with each receptive field:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} f_i \omega_i = \sum_{i=1}^{N} f_i R_i(\boldsymbol{x}) \tag{1.6}$$

where f_i is the scalar value, or strength, of the i^{th} receptive field and N is the number of receptive field units. Such RBF network is able to produce the normalized response function as the weighted average of the firing level as shown in equation (1.7).

$$f(\boldsymbol{x}) = \frac{\sum_{i=1}^{N} f_i \omega_i}{\sum_{i=1}^{N} \omega_i} = \frac{\sum_{i=1}^{N} f_i R_i(\boldsymbol{x})}{\sum_{i=1}^{N} R_i(\boldsymbol{x})} \tag{1.7}$$

Since f_i is linearly weighted to represent the output, i.e. a linear relationship exists between the weights and the network output; there are a number of advantages to this structure. The optimisation of the parameters based on standard regression techniques and methods such as regulation, experiment design and recursive estimation are easier to transfer to such structures than to MLP networks. For example, a standard linear parameter estimation scheme can be employed to train the weights with the ordinary linear least squares algorithm (Moody and Darken, 1989).

This linear learning procedure for RBF networks is much easier to apply than the complicated nonlinear learning procedure for MLP networks. With only one hidden layer, the output layer becomes a linear combination of the hidden layer signals. So, the RBF network allows for a much simpler weight updating procedure and subsequently opens up greater possibilities for stability proofs and network robustness, as the network can be described readily by a set of linear equations.

1.2 Neuro-control Approaches

Neural networks have been used for different purposes in the context of control. Their fast parallel computation and their use of generic function forms for complex non-linear mapping have motivated the proposal of a growing number of control schemes involving neural networks (Hunt et al., 1992, Irwin et al., 1995, Agarwal, 1997, *Nøgaard* et al., 2001). Generally, most of the existing methods used to develop a neuro-controller are based on inverse control. This section historically introduces neuro-control techniques based on the neural network model and analyses the advantages and disadvantages of these techniques.

1.2.1 Neuro-control Techniques Based on System Inverse

When neural networks originally were proposed for controlling unknown nonlinear systems, one of the first methods reported was the training of a neural network to act as the inverse of the system and the use of the inverse network as a controller. The nonlinear internal model control (NIML), which was proposed by Bhat and McAvoy (1990), Hunt and Sbarbaro (1991) and Psichigios and Ungar (1990) is one example. This is formulated in equation (1.8),

$$y(k+1) = g[y(k),\cdots,y(k-n+1),u(k),\cdots,u(k-m)] \qquad (1.8)$$

where *y(k)* is system output, *n* is system order, *u(t)* is the control input, *m* is input order and $g(\cdot)$ is a system function that describes the relationship between the inputs and outputs. The desired network is then the one that isolates the most recent control input *u(k),*

$$\hat{u}(k) = \hat{g}^{-1}[y(k+1), y(k)\cdots y(k-n+1),u(k),\cdots,u(k-m)] \qquad (1.9)$$

where $\hat{u}(k)$ is the computed control signal and $\hat{g}(\cdot)$ is the neural network function. Assuming such a network has been obtained, it can be used for controlling the system by substituting *y(k+1)* with the desired output, the reference signal *r(k+1).*

The first such approach for developing a neuro-controller was by (Widrow and Smith, 1964), who tried to replicate a human-expert-based controller for the inverted pendulum. This was used for plants controlled by humans, for which it is difficult to design a standard controller (Hunt et al., 1992). The neural training consists simply of learning the mapping between the sensory information received by the human controller and the control output.

A similar approach involves training a neuro-controller to mimic the behaviour of another controller designed by a conventional method (Ruano et al., 1992, Yu and Lu, 1998). In most of the applications, the neuro-controllers are trained to become a system inverse (Gomi and Kawato, 1993, Lightbody and Irwin, 1997). This approach is especially useful for plants having fast dynamics. The following sections will describe some techniques proposed for establishing the inverse model: the direct inverse training method, the specialised inverse training method and the most popular back-propagation method. However, if the inverse model is unstable (corresponding to zeros outside the unit circle in the discrete time linear case), the closed-loop system becomes unstable. Unfortunately, this situation occurs quite frequently in practice (Åström et al, 1984).

1.2.1.1 Direct Inverse Neuro-control Training

The most straightforward way of training a network as the inverse of a system is to receive the system output as an input and the system input as an output. The model is trained by inputting to the controller a sequence of signals *u(k)*. Some examples can be found in (Widrow and Stearns, 1985). Note that although there is a feedback path between the system output y and the input vector $\boldsymbol{x}$, this control system is open loop in the sense that it does not take into consideration the error e between the system output and the desired output, r, to decide a suitable control action. Clearly, there will be no action taken by the neural network to compensate for any deviation between the system output and the desired output. Hence, in that respect, this scheme is not a feedback control system but an open loop one.

The main difficulty in applying this training method is to choose the training signal *u(k)*. The system must be brought into the desired controller operating region. This is difficult to achieve without strong a priori knowledge about the system.

1.2.1.2 Specialised Inverse Neuro-control Training

Specialised training (Psaltis et al. 1988) is a goal directed online neuro-control approach. This feature is the fundamental difference between the specialised inverse training method and direct inverse training method. Inspired by the recursive training algorithms (e.g. recursive least squares algorithm), the network is trained to minimize the criterion J_k, which is described as below:

$$J_k\left(\theta, Z^k\right) = J_{k-1}\left(\theta, Z^{k-1}\right) + \left[r(k) - y(k)\right]^2 \tag{1.10}$$

With $Z^k = [y(k), \cdots y(k-n+1), r(k), r(k-1), \cdots r(k-m)]$ and that θ is the coefficient of Z^k. Based on this criterion, the Jacobian of the process is necessary (Psaltis et al., 1988). Since specialised training is an on-line approach, the combination of the need to adjust weights and the slow convergence of a gradient method may cause serious consequences. It is often highly recommended to use the direct inverse training approach to provide an adequate initialisation for the network and the specialised inverse neuro-control training is only used for "fine-tuning" of the controller (*Nørgaard* et al., 2001).

Back-propagation through time neuro-control learning

To avoid the problem associated with the specialised inverse control training, a number of researchers (Nguyen and Widrow, 1990, Narendra and Parthasarathy, 1990) have independently developed the back-propagation through time neuro-control learning method. The problem of the specialised inverse control training method is that the performance error $e_y = r - y$ is not reliable because it is not directly related to the neuro-control output, *u*. To train the controller, we need to know the error in the controller output. Therefore (Nguyen and Widrow, 1990, Narendra and Parthasarathy, 1990) propose to emulate e_u, a "virtual" error on the output of the controller:

$$e_u(k) = \frac{\partial \hat{y}(k)}{\partial u(k-1)} e_y(k) \tag{1.11}$$

where $\hat{y}(k)$ is the observed system output. In this way, the error e_y, back-propagated toward the input layer of the feed-forward model, should correspond to the error e_u (assuming the input, *u*, has a linear effect on the system). This makes the control training feasible in many cases.

Narendra and Parthasarathy (Narendra and Parthasarathy, 1990) introduce a reference model into the training structure, which gives a desired transient system output $\bar{y}$ rather than a simple fixed reference output ***r***. However, this also leads to a system inverse based neuro-controller since it is the desired output $\bar{y}$ that drives the neuro-controller.

In summary, neuro-control methods based on the system inverse have serious drawbacks. The most important concerns the lack of feedback. It is not trivial to learn the inverse dynamics of highly nonlinear systems. Furthermore, not all systems have an inverse model. However, considering the modelling approximation ability of neural networks, the model of the plant dynamics could be used to predict future states of the plant. The model based predictive controller (MBPC) is then worth to consider. This algorithm is presented in the next section while highlighting the usefulness of neural networks in this context.

1.2.2 Neuro-model Based Predictive Control

Model Based Predictive Control (MBPC) is a criterion-based control algorithm, which uses an optimiser to solve for the control trajectory over a future time horizon based on a dynamic model of the process. It has become a standard control technique in the process industry over the past two decades.

A prime characteristic that distinguishes predictive controllers from other controllers is the idea of a receding horizon; at each sample, the control signal is determined, so as to achieve a desired behaviour in the following H_C (control horizon) time steps. This idea is also appealing because it relates to many of the control tasks that one, as a human being, carries out on a daily basis. This intuitive foundation can, to some extent, accommodate the tuning of the design parameters.

Predictive control can be used together with instantaneous linearisation in an implementation similar to the approximate minimum variance design (Chen et al., 1990). However, predictive control algorithms apply to a full nonlinear model of the system as well (Wills et al., 1992). Due to its flexibility, predictive control can handle a large class of systems. Here, the Generalised Predictive Control (GPC) method is taken as an example. It was originally proposed by Clarke et al. (1987a, 1987b) as an alternative to pole placement and minimum variance designs used in self-tuning regulators. Although it originated in an adaptive control context, GPC has many attractive features, which definitely makes it worthwhile considering even for time-invariant systems.

The idea behind generalized predictive control is to minimize a criterion of the following type

$$J(N_1, N_2) = E\left\{ \sum_{j=H_m}^{H_p} [y(k+j) - w(k+j)]^2 + \sum_{j=1}^{H_c} \lambda(j)[\Delta u(k+j-1)]^2 \right\} \quad (1.12)$$

where H_P and H_m are the minimum and the maximum costing horizon; H_C is the control horizon; $\lambda(j)$ is a control weighting sequence, which is introduced to make a trade-off between the minimization of tracking error and the minimization of the controller output and to penalise excessive changes in the manipulated input; *y, w, u* are the controlled output, desired output and manipulated control input, respectively.

To minimize the criterion, when the prediction are nonlinear in the control inputs, is a quite intricate optimisation problem. In order to determine the minimum, it is necessary to apply an iterative search method similar to the strategies used when training neural networks. For linear systems, the optimisation problem can be solved beforehand, which will be given in detail in section 3.3.3. However, for nonlinear systems, the optimisation problem must be solved at each sample, relating to a sequence of future control inputs. From this sequence, the first component *u(k)*, is then applied to the system. Generally, solving the optimisation problem can be done in two ways.

- The first idea is to derive the most suitable optimisation algorithms for minimization of the MPC criterion by successive recursion of a deterministic neural network model. This makes calculation of the control input very difficult, as it has to be determined with an iterative minimization algorithm deployed at each sample (Draeger et al., 1995, Schenker and Agarwal, 1998). Such an approach is computationally intensive.

- The second idea is based on an approximate minimum variance predictor derived from the instantaneous linearisation of a neural network autoregressive exogenous (NNARX) model. This makes the calculation of control inputs much simpler as the minimum of the controllers is unique and easy to find. However, a problem may occur if the approximate models obtained from instantaneous linearisation are accurate only in a very narrow region of the current operating point. Meanwhile, the linearisation may suffer from sensitivity to noise (Sales and Billings, 1990, Geng and Geary, 1997).

1.2.3 Remarks on Neuro-control

As discussed during the chapter, the accuracy of neural network models offers a significant improvement over linear models. However, process control engineers are faced with the difficulties resulting from neural network models' black-box presentations of the nonlinear dynamic properties, while the widely used multi-layer perceptron (MLP) approach has important drawbacks. Firstly, and most importantly, due to the neural network model's unclear representation of the system, it is not possible (or it is too difficult) to investigate their modelling and control properties. The issue of controllability is seldom considered, although some work has been done under significant practical restrictions (Narendra, 1996). This issue is very constraining, since it is often necessary to investigate the controller properties to make sure that it satisfies stability and robustness criteria. Moreover, the back-propagation learning algorithm is essentially a steepest-decent gradient algorithm and can either converge on a local minimum or be extremely slow to converge. There is no guaranteed solution even given the required model structure, due to the existence of flat areas and/or local minima in the error surface (Rumulart and McClelland, 1986).

Hence, even if the described neuro-control training methods, especially the MPC, are promising approaches, mainly due to their non-linear control capability, their applications are highly constrained by the drawbacks inherent in neural networks, for the control signal u is required to appear linearly in the MPC equation.

This may explain why, from the early 1990's, there is a rapid increase of research interest in the application of operating regime based multiple models in control. The LM network is one type of these multiple models, which use neuron supported functions that are active over some local regimes of the system operating range to combine the local models. The advantages of this approach are purported to be their simple approximation of the nonlincar property and their insight into the global dynamics from local models (Johansen and Foss, 1995, Murray-Smith and Johansen, 1997, McLoone et al., 2001).

1.3 Multiple Model Networks

The last decade has shown a rapid increase in the use of local model representations of non-linear dynamic systems. Control techniques based on multiple models have been applied extensively in recent years. Numerous applications have been reported in the mechanical, chemical, aeronautical industries and many other fields (Boskovic and Mehra, 1999,).

Control techniques based on multiple model methods include a number of approaches: Takagi-Sugeno (1985) fuzzy systems, local model networks (Johansen and Foss 1993), gain-scheduled control (Shamma and Athans, 1990), local dynamic modelling with self-organising maps (SOM) (Principe et al, 1998), the smooth threshold autoregressive (STAR) models of Tong and Lim (1980) and the state dependent models of Priestly (1988). Most commonly, linear local models are employed in these approaches for their simplicity and continuity to the well-developed control methods and theories for linear systems. Then the weighted sum of the local sub-models provides a qualitative high-level description of the nonlinear system. The model parameters can be obtained from prior knowledge, linearisation of a physical model or identified from measured data.

The construction of local model networks for interpolating the behaviour of locally valid models offers an attractive and intuitively pleasing method of modelling non-linear systems. The advantage of this approach is purported to be its insight into global dynamics from the local models. In addition, it is well-known that every model of the systems is based on some simplifying assumptions, and that different models maybe useful for understanding different aspects of the same phenomenon. Multiple models are needed to describe the process characteristic when they change intensively with time, as for example, chemical processes (Li et al. 2001, Zhang and Morris, 2001), paper mill (Chow et al., 1996) and flight control (Boskvoic and Mebra, 1999). This is another source of motivation for studying multiple model systems.

In terms of control, the local model network potentially provides a convenient framework for obtaining both stability and improved performance simultaneously for the control of nonlinear systems. Control techniques based on interpolation between different controllers have been applied extensively in recent years. Correspondently, the issue of stability has become one attractive topic as well. Many publications have been given for fuzzy controllers by using polytopic (Farinwata et al., 2000, Filev 1996) and Lyapunov functions (Wang et al., 1996, Daaffouz et al, 2002). Considering the functional similarities (see section 2.3.4), some techniques applied for fuzzy systems may be employed for the development of LC (local controller) network.

1.4 Objectives of Thesis

Despite the wide range of methodologies and applications of multiple models in nonlinear systems modelling and the growing interest in this area, controller design and analysis based on multiple models is still at an early research stage. The thesis will focus on local model network applications in control.

The thesis will investigate controller design approaches for the control of complex nonlinear dynamical systems. It aims to systematically develop an overall global controller, with guaranteed stability and desired performance based on multiple models, from the practical application point of view. Specifically, the research will concentrate on the issues of controller design based on conventional local model networks and velocity-based multiple model networks.

Meanwhile, the thesis will provide a systematic presentation of features, advantages and problems encountered, and available solutions, in the application of multiple models in control. This presentation will include the main theoretical results and design procedures with the aim of providing a critical overview of the field, and a useful entry point into relevant literature.

1.5 Contributions of Thesis

Original work contained in this thesis include the following topics:

1. The development and application of gain-scheduled local controller networks by employing the PID controller and the generalised predictive control (GPC) approach, based on affine local model (ALM) networks in the control of the Continuous Stirred tank Reactor (CSTR) process (Chapter three).

2. Application of Linear Parameter Varying (LPV) based GPC to the CSTR process (Chapter three).

3. The development and application of a state feedback integral controller based on velocity-based multiple model networks for the control of the CSTR (chapter five).

Some minor contributions in the thesis includes

1. A local model network application, with property analysis in the modelling of a continuous stirred tank reactor (CSTR) simulated process (chapter two).

2. Application of stability analysis and control synthesis to affine LM (Local Model)/LC (Local Control) networks based on the state feedback control approach (chapter four).

3. An application of the state observer in the control of the CSTR simulated process based on velocity-based multiple model networks; a proposal for the control synthesis of the velocity-based approach (chapter five).

4. The development of the discrete-time version of velocity-based multiple model networks (chapter five).

1.6 Outline of Thesis

The thesis will deal with the controller design issues for nonlinear systems based on conventional LM networks and velocity-based multiple model networks.

Chapter two will introduce how to structure a conventional LM network, in which the choices of local model structure, validity functions and interpolation schemes, respectively, will be investigated. It will also look into the properties of both linear local model (LLM) and affine local model (ALM) networks for the modelling of nonlinear systems. Simulation work will be carried out on a highly nonlinear simulated process, the continuous stirred tank reactor (CSTR).

Chapter three will review the reported contributions in controller design based on multiple models and present how to formulate the gain-scheduled local controller network and linear parameter varying (LPV) controller based on local model networks. Generalize predictive control (GPC) will be proposed as the basic technique for the controller design mentioned above under the assumption that system changes 'slowly' enough. Simulation work will be carried out in the application of the algorithms to the control of the CSTR simulated process, followed by analysis of results.

Chapter four will consider the stability conditions for blending systems with affine local models, in which the offset term is nonlinear and dependent on parameters that formulate interpolation functions. By employing Quadratic Lyapunov functions, theories will be developed to determine the bound for open loop blending systems. Then a state feedback controller in the LC network framework will be proposed with the aim of providing an overall stable closed-loop system with guaranteed stability and the bound for the compensated system will be determined.

Chapter five will consider two issues. The first issue is the controller design based on velocity-based (VB) multiple model networks. A novel controller will be developed based on the VB local model networks by employing integral state feedback control. Simulation work will be carried out in the application of the control algorithm to the CSTR simulated process; a comparison of results with those of the gain-scheduled LC networks proposed in chapter 3 will be made. In addition, a discrete-time version of the network will be developed based on the continuous time version. Modelling properties will be investigated by simulation work carried out on the CSTR simulated process.

Chapter 6 concludes the thesis and outlines some future research directions.

Chapter 2

Conventional Local Model Networks

2.1 Introduction

In the design of nonlinear control systems, to achieve good performance (for example, speed and accuracy) and robustness, it is desirable to derive an accurate and meaningful model of a given practical plant, i.e. a model that captures the key dynamics of the plant in the operational regime of interest and provides transparent insight into the nonlinear systems.

In recent years, an increasing interest in the use of local model representations of nonlinear dynamic systems has been reported in the literature (Townsend and Irwin, 1999, Narendra and Cheng, 2000, Principe et al, 1998, Sharma et al, 2002). The idea started from partitioning a complex problem into a set of simple sub-problems, whose solutions are combined together through interpolation schemes (also called scheduling, blending and clustering schemes) to give a global solution to the original problem. If multiple models are used for nonlinear process modelling, techniques for multiple model development need to be formulated and in turn, the following issues should be of concern:

- Decomposition of local regimes from the full operating regime of the nonlinear system

- The structure selection and parameter estimation of the local models
- Interpolation schemes to combine the local models

These three issues are highly correlated. The basic structure of multiple modelling includes a number of approaches: gain-scheduled control (Shamma and Athans, 1990), fuzzy inference systems (Takagi and Sugeno, 1985), local model (LM) networks (Johansen and Foss, 1993), local dynamic modelling with self-organising maps (SOM) (Principe et al, 1998), and adaptive control of multiple models (Narendra and Balakrishan, 1994a). They are different from each other in the methods used for the interpolation stage. Among these techniques, LM networks and SOM networks employ neural networks in combining the local models to structure the global nonlinear model. The LM network applies a Radial basis function (RBF) network and the SOM network applies Kohonen neural networks. Conventional gain scheduling control is done in an ad-hoc manner by the use of heuristic guidelines (Nichols et al, 1993) and additional information about the plant (Rugh, 1991). Adaptive multiple model control is a kind of piece-wise linear control (without interpolation). It leads to 'switching' and 'tuning' techniques for the control of complex nonlinear systems. The former takes rapid measures to avoid catastrophic failures, while the latter involves more leisurely steps taken to improve the control performance of the nonlinear system. The fuzzy inference system structure is very close to the LM network structure under some minor assumptions; a functional equivalence analysis between them will be given in section 2.3.4.

The research work focuses on the application of LM networks in the control of nonlinear dynamic systems. LM networks aim to use a small number of relatively simple local models to approximate a wide range of the operating regime of a complex nonlinear system. The operating regime plays an important role in this area. The basis of the LM network consists of decomposing the full range of the system operating regime into a number of overlapped local operating regimes, in which a simple local model (linear or affine model) describes the local dynamics. Then these local models are smoothly combined in a neural network structure to yield a global model. The model parameters can be obtained from prior knowledge, linearization of a physical model or identified from measured data sets. The advantages of this approach are purported to be its simplicity and the insight into global dynamics from the local models (McLoone et al, 2001). Furthermore, because linear models are most commonly selected to describe local dynamics, the well-developed control theory for linear systems can be extended in some simple ways to the control of highly nonlinear systems.

This chapter will begin by presenting a detailed introduction on structuring a conventional LM network, in which the choices of local model structure, validity functions and interpolation schemes, respectively, will be investigated. Then, the properties of the LM network are discussed and analysis shows that the LM network is functionally equivalent to popular fuzzy inference systems. Next, simulation work is carried out on a simulated process, the continuous stirred tank reactor (CSTR), which is a highly nonlinear plant. Finally, concluding remarks are provided.

2.2 Architecture of Conventional LM Networks

Operating regime based LM networks are constructed by decomposing the system into a number of smoothly over-lapping local operating regions in which a simple local model describes the local dynamics. The global nonlinear model is created by using the representational ability of the radial basis function (RBF) network to form the LM network structure, which is able to apply the advantage of RBF in learning compared with MLP approaches, and allows more freedom in the selection of the scalar value, f_i, which in LM networks denotes a local model function. Compared with neural network models, LM networks offer several advantages:

(1) The LM network develops a relatively clear representation of nonlinear processes since each model is linear. This enables easy analysis of overall properties and makes the conversion of the LM network into a local controller network (LCN) straightforward (Gawthrop, 1995, 1996). The LM network is in this respect one of the few nonlinear modelling approaches oriented for control design.

(2) The learning of local models is quick, computationally efficient and straightforward since linear regression methods can be applied.

(3) It is possible to introduce a priori knowledge in LM networks by using different sorts of local models and by determining the operating regions of each model.

(4) The system order is the main information required in applying a LM network (Johansen & Foss, 1995, Nelles, 1997), which allows a weakening of a priori knowledge needs for the system, although good knowledge about the system will help in the structuring of the LM network.

These advantages highlight the powerful potential of LM networks as a nonlinear modelling and control approach. In addition, there have already been theoretical supports available that LM network is able to approximate any continuous function given a sufficient number of local models since the early 1990s (Funahashi 1989, Johansen and Foss, 1993).

The following sub-sections investigate the basic problems in relation to structuring LM networks, which include

- How to decompose the overall model to a set of operating regimes?
- How to define the proper operating regime based on a priori knowledge of the system?
- How to optimize the number and parameters of local models? and especially
- How to optimize the LM network towards control objectives?

2.2.1 Operating Regime Decomposition

Any model has a limited range of operating conditions in which it is sufficiently accurate to serve its purpose. This range may be restricted by several factors, such as validity of linearisation, modelling assumptions, stability properties or experimental conditions. A model that is useful in a relatively small region rather than the full range of operating space is called a local model, as opposed to a global model, which is effective over the full range of operating space. LM network consists of a set of local models, each describing different dynamics of the same system but at different operating regions. The output of these local models are weighted and summed to give the global model output.

The general architecture of LM networks for a single-output nonlinear dynamical system is shown in figure 2-1, and it can be described by

$$\hat{y}(t) = \sum_{i=1}^{N} \rho_i(\psi(t))\hat{f}_i(x) \text{ , } \psi \in Z \tag{2.1}$$

where the operating space Z, a subspace of the input space, is decomposed into N regimes:

$$Z = \cup Z_i \text{, } i=1,\ldots,N \tag{2.2}$$

In equation (2.1), $\hat{y}(t)$ is the LM network output at time t, x represents the model input vector and $\hat{f}_i(x)$ defines the local model for the i^{th} regime. The associated validity functions (or basis functions) for the i^{th} regime is given by $\rho_i(\psi(t))$, which is a function of the scheduling vector $\psi(t)$.

The local models are typically linear or affine, but they could also be nonlinear. N is the number of validity functions. To an extent, defined by the activation of the validity function ρ_i, the output $\hat{f}_i$ of the corresponding local model defines the model output $\hat{y}$.

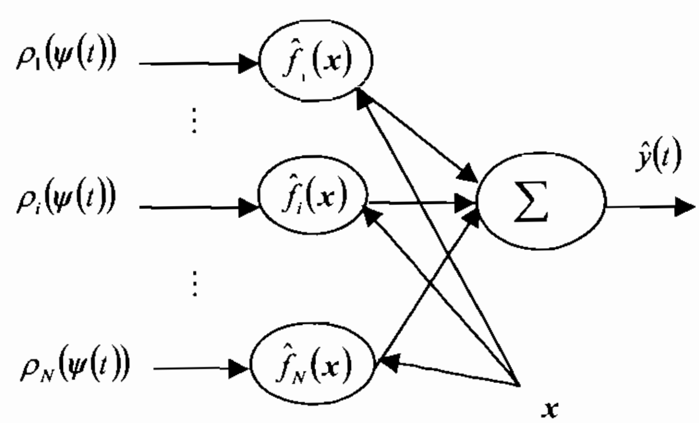


Figure 2-1. LM network structure

Comparing the LM network with the RBF network introduced in section 1.1.2, it can be seen that the LM network can be taken as a generalised RBF network, assuming the input to the RBF network is defined as a scheduling vector $\psi(t)$ and the weighted sum of the functions f_i (the scalar value of i^{th} receptive field in the RBF network) is defined as $\hat{f}_i(x)$, which describe the same nonlinear system at different operating regimes.

In other words, the LM network structure has the advantages inherent in the local nature of the RBF network; the learning algorithms and the theorems on the representational ability for the RBF network can be applied to the learning of the LM network. At the same time, because of the more powerful local models associated with the basis functions, the LM network does not require as many basis functions as the RBF network to achieve the desired modelling accuracy.

2.2.2 Locally Valid Models

Theoretically, the local model structure can employ either linear (or affine) or nonlinear models. However, from a practical point of view, local models will be commonly chosen as linear (or affine) models (Murray-Smith, 1994) for computational simplicity and convenience. They can be chosen from the linearisation of a nonlinear system around some operating points corresponding to the 'centres' of given sub-regimes, if the physical model has been defined.

2.2.2.1 State Space Model

Consider the general nonlinear state space SISO (single input, single output) system, with state vector x and input u:

$$\dot{x}(t) = f(x(t), u(t))$$
$$y(t) = g(x(t), u(t)) \tag{2.3}$$

where $\boldsymbol{x}$ denotes the state vector, u is the system input. It is assumed that $y = \boldsymbol{C}\boldsymbol{x}$, without loss of generality, because it is common and practical to define system output y proportional to the state vector.

In many cases, the behaviour of the nonlinear system near an operating point $(\boldsymbol{x}_0, u_0)$ can be described by a linear time-invariant system. To see this, we consider state and input trajectories that are small perturbations away from the operating point:

$$\boldsymbol{x}(t) = \boldsymbol{x}_0 + \delta\boldsymbol{x}(t)$$

$$u(t) = u_0 + \delta u(t) \tag{2.4}$$

where u_0 is the nominal input and $\delta u(t)$ is the perturbation input. The input and state vector obey the differential equation, determined by submitting (2.4) into (2.3):

$$\delta\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}_0 + \delta\boldsymbol{x}(t), u_0 + \delta u(t)) \tag{2.5}$$

Expanding the right-hand side of (2.5) in a Taylor series about ($\boldsymbol{x}_0$, u_0)

$$\delta\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}_0, u_0) + \frac{\partial f}{\partial \boldsymbol{x}}\Big|_{(\boldsymbol{x}_0, u_0)} \partial\boldsymbol{x}(t) + \frac{\partial f}{\partial u}\Big|_{(\boldsymbol{x}_0, u_0)} \partial u(t) + h_0(o) \tag{2.6}$$

in which $h_0(o)$ is the high order error of the Taylor series. If defining $\boldsymbol{A}_0 = \frac{\partial f}{\partial \boldsymbol{x}}\Big|_{(\boldsymbol{x}_0, u_0)}$, $\boldsymbol{B}_0 = \frac{\partial f}{\partial u}\Big|_{(\boldsymbol{x}_0, u_0)}$ and substituting (2.3) into (2.5) gives

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_0\boldsymbol{x}(t) + \boldsymbol{B}_0 u(t) + \boldsymbol{\alpha}_0 + h_0(o) \tag{2.7}$$

in which $\boldsymbol{\alpha}_0 = f(\boldsymbol{x}_0, u_0) - (\boldsymbol{A}_0\boldsymbol{x}_0 + \boldsymbol{B}_0 u_0)$.

If $h_0(o)$ is ignored, at an equilibrium point, the constant trend term $\boldsymbol{\alpha}_0$ vanishes to zero and the dynamics are fully captured by the $(\boldsymbol{A}_0, \boldsymbol{B}_0)$ parameters, which lead to the normal linear local model (LLM). The structure of (2.6) contains extra degrees of freedom $\boldsymbol{\alpha}_0$, which leads to a reasonable approximation in a small neighbour-hood of the linearized point. This structure formulates an affine local model (ALM). The properties of both LLM and ALM will be investigated in sections 2.3.2 and 2.3.3, respectively.

2.2.2.2 Input/output Model

If the local models are of transfer function form, we consider a discrete-time nonlinear SISO system having the general form

$$y(k) = f\left(y(k-1),\ldots,y(k-n_y),u(k-n_k),\ldots u(k-n_k-n_u)\right)+e(k) \qquad (2.8)$$

where *y(k)* denotes the system output, n_y denotes the system order, *u(k)* denotes the system input, n_u denotes the system input order, n_k denotes the time delay and *e(k)* denotes a zero-mean disturbance term. Attention is restricted to single-input, single-output systems. Defining the information vector as

$$\phi(k-1) = [y(k-1),\ldots y(k-n_y),u(k-n_k),\ldots,u(k-n_k-n_u)]^T \qquad (2.9)$$

then equation (2.8) can be written as

$$y(k) = f(\phi(k-1)) + e(k) \qquad (2.10)$$

The aim of the LM networks approach is to find a parameterized structure, which approximates the nonlinear function *f*.

Consider a linearisation of equation (2.7) about a nominal operating point (e.g., by taking a first-order Taylor series expansion), which results in a linear autoregressive model with exogenous input (ARX) structure:

$$y(k) = \phi_{ar}^T(k-1)\boldsymbol{\theta} + e(k) \qquad (2.11)$$

where

$$\phi_{ar}^T(k-1) = [-y(k-1),\ldots,-y(k-n_y),u(k-n_k),\ldots u(k-n_k-n_u)]^T \qquad (2.12)$$

and

$$\boldsymbol{\theta} = [a_1,\ldots,a_{n_y},b_0,\ldots,b_{n_u}]^T \qquad (2.13)$$

is a constant parameter vector. Note that in equations (2.11)-(2.13), all signals now strictly represent deviations from the nominal operating point. With the definitions in equation (2.14) and (2.15), the linear system (2.11) can be written in the familiar transfer function form

$$y(k) = \frac{q^{-k}B(q^{-1})}{A(q^{-1})}u(k) + e(k) \tag{2.14}$$

Here q^{-1} is the unit delay operator and the polynomials A and B are

$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_{n_y} q^{-n_y}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \ldots + b_{n_u} q^{-n_u} \tag{2.15}$$

In sub-section 2.2.2.1 and 2.2.2.2, locally valid models were introduced in both state space and input-output transfer function form. In later chapters, depending on the purpose of the modelling and the approaches used for controller design and property analysis (such as robustness, stability and sensitivity analysis), both state space and input-output representations will be applied.

2.2.3 Choices of Validity Functions

Having partitioned the full operating range into a number of simpler regimes and built local models within each operating regime, one natural question that appears is how to combine these sub-models to achieve a global model. In other words, how does the interpolation scheme decide when, where and how to 'switch' between the local models. Validity functions play significant roles in interpolation between the local models. They smooth the 'switching' among the local models, improve the global approximation accuracy and balance the trade-off between local performance and global performance. A validity function should basically satisfy the following properties:

- A validity function limits its outputs to a value between 0 and 1; the activation of a validity function decreases with increasing distance of the input from its centre. The activation fades to zero for inputs that are far from the centre.

- All validity functions form a partition of unity of their input space, that is $\sum_{i=1}^{N}\rho_i(\psi)=1, \psi \in Z$; this ensures that every single point in the input space is covered to the same degree.
- The shape of the validity functions is smooth.

Generally, the Gaussian bell function is the validity function used most widely. Other popular validity functions are B splines (Kalvi, 1993), multivariate adaptive regression splines (MARS) (Friedman, 1991, Pischogios et al., 1992), and the kernel basis functions (KBFS) (Hlavackova, 1996).

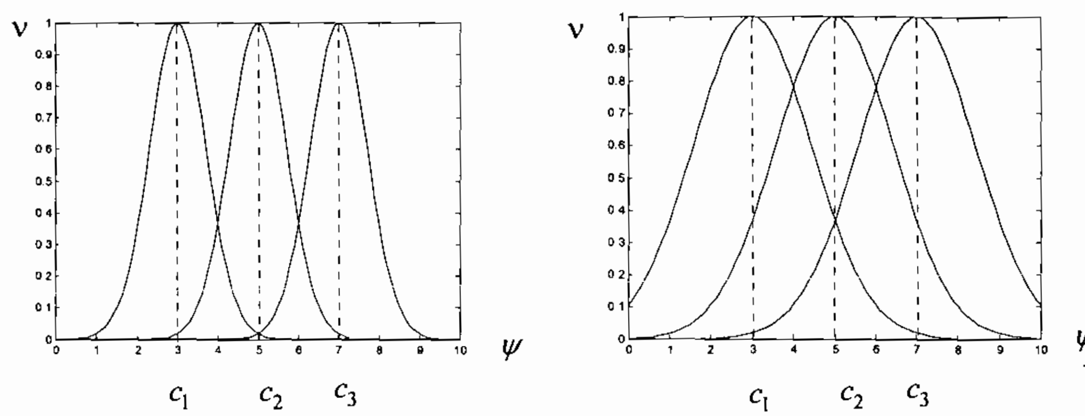
Here, the Gaussian bell function is taken as an example. A general definition of the radial Gaussian bell function may be given by

$$v_i(\psi, c_i, \sigma_i) = \exp\left(-\frac{(\psi - c_i)^T(\psi - c_i)}{\sigma_i^2}\right) \tag{2.16}$$

Here, each basis function has two parameters: the centre vector c_i and the width factor σ_i. If Gaussian bell functions are used as validity functions, their width can be adjusted according to the mean distance to the n nearest neighbours. For the radial Gaussian bell functions described, the width σ_i can be set as

$$\sigma_i = k_s \frac{1}{n}\sum_{j=1}^{n}\left|c_i - c_{i,j}\right| \tag{2.17}$$

where c_i is the current centre and $c_{i,j}$ is the centre of the j^{th} nearest neighbour to c_i. The scaling factor k_s defines the degree of overlap between the validity functions and has the same value for all centres. Figure 2-5 shows how the overlapping area changes among three Gaussian functions centred at $c_1 \sim c_3$, which are equal to 3, 5 and 7 respectively, when the scaling vector k_s changes from 1 to 2. The overlapping area increased significantly comparing Figures 2-2 (a) and 2-2(b).



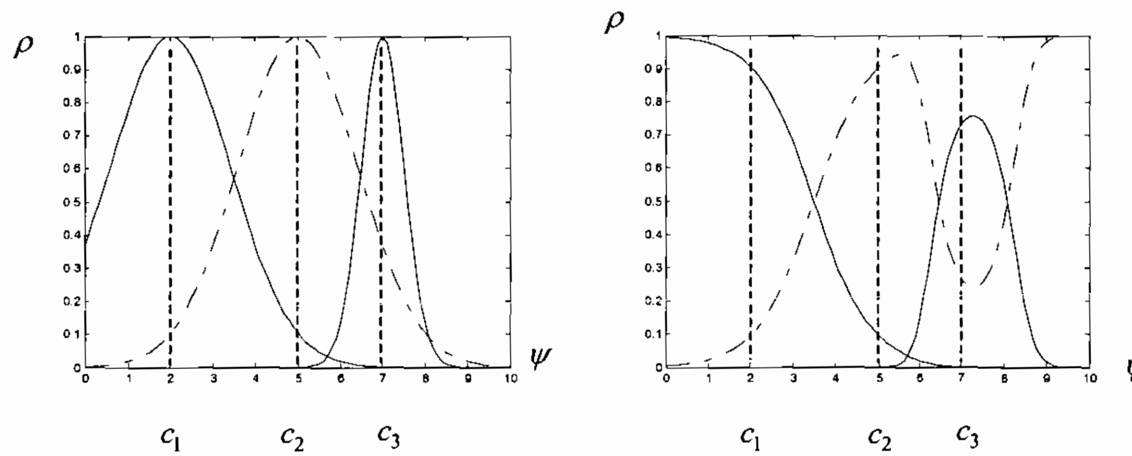
(a) k_s=1 (b) k_s=2

Figure 2-2. Overlapping of Gaussian bell functions

In practice, the Gaussian bell functions are normally normalised to unity:

$$\rho_i(\psi, c_i, \sigma_i) = \frac{\nu_i(\psi, c_i, \sigma_i)}{\sum_{i=1}^{N} \nu_j(\psi, c_i, \sigma_i)} \tag{2.18}$$

In many cases, the use of normalised basis functions has resulted in a structure which can be less sensitive to poor centre selection. This also ensures every point in the operating space is covered by the basis function to the same degree i.e. the basis functions form a partition of unity across the input. Meanwhile, the area outside of the designed operating space is also covered by the normalised basis functions. However, the normalisation of basis functions also brings some side effects, for example, the change in shape of the basis functions and the possible loss in smoothness of the representation. These problems could cause the loss of the independence of local models, and lead to unpredictable (even unstable) behaviour in dynamic models (Shorten and Murray-Smith, 1994). Figure 2-3 shows an example of some of problems associated with normalisation.



(a) Un-normalised validity function (b) Normalised validity function

Figure 2-3. Normalisation effect on validity function

Three Gaussian functions (G1~G3) are centred at $c_1 \sim c_3$, which are equal to 2,5 and 7 respectively; the scaling vectors are defined as $k_{s1} = 2$, $k_{s2} = 2$ and $k_{s3} = 1$. The dashed vertical line highlights where the centres lie. Figure 2-3 (a) shows their overlapping operating area focus on $\phi = 2 \sim 5$ between G1 and G2 and $\phi = 5 \sim 7$ between G2 and G3. However, after normalisation, as shown in Figure 2-3 (b), G2 plays an important role in both its own working area and the G3 working area, which invades the independence of G3. Secondly, the normalisation results cover the whole of input space rather than the specified space defined by the training data. One more issue is that the maximums of the Gaussian bell functions may move away from their centres.

After normalization, all the validity functions preserve the order of their relative validity, i.e. if $v_i \geq v_j$ then $\rho_i \geq \rho_j$, but their firing levels are not proportional to the validity functions. To get a proper LM network targeted for modelling or control objectives, it is important not to normalise without the attention to the system, but to compensate for the normalisation by altering the design criteria (i.e. centre positions and width magnitude) for the structure identification procedure based on a priori knowledge of the system. The solution of this problem concerns the interpolation technology, which involves the target directed learning of the LM networks.

2.2.4 Interpolation

The main issues of LM networks implementation concern the identification of the structure (number of local models and partition patterns) and the parameters (validity functions and local models). These tasks are interlinked through interpolation technology by using validity functions. The role of interpolation is to provide smooth 'transition', in some sense, between the local models, with the aim of achieving an ultimate global representation of complex nonlinear systems with only a small number of local models. Therefore, the interpolation methods are of significance to the utility of the LM network.

One major concern in interpolation is how well the multiple models can approximate a real system, which depends on the formulation of the learning task for LM networks. Ideally, it is desirable that a LM network should give accurate global nonlinear prediction and, at the same time, that its local models are close approximations to the local linearization of the nonlinear dynamic system at the operation point. The former requirement is of significance in the global performance of the modelling; while the latter requirement is important in many applications where the local models are used individually to aid validation and interpretation of the models. These requirements define a multi-objective identification problem, namely, the construction of a dynamic model that is a good approximation of both local and global dynamics of the underlying system. However, these objectives are normally conflicting. From the control point of view, the solution to the conflict depends on the control objective and the controller design approaches.

Most existing learning algorithms choose the parameters to minimize a global mean square error (MSE) objective function

$$J_G = \frac{1}{M}\sum_{k=1}^{M}\left[d(k) - \hat{y}(k)\right]^2 \qquad (2.19)$$

where $d(k)$ is the output of the real system (or expected output), $\hat{y}(k)$ is the output of the identified model network defined in equation (2.1), and M is the length of training data set. The objective of learning is to try to adapt local model and interpolation function structure and parameters to minimize the criterion function related to the deviation of the model output from the desired system output.

Solving this minimization problem involves the estimation of validity function $\rho_i(\psi(t))$ parameters and the estimation of local dynamic model $\hat{f}_i(x)$ parameters. Normally, the local model parameters represent the linear behaviour of the LM network and thus can be learned relatively easily by some non-iterative linear algebraic methods, such as the least squares method. In contrast, the validity function parameters represent the location and the shape of the interpolation functions that influence, in a nonlinear way, the LM network behaviour. Therefore, iterative nonlinear optimisation techniques such as steepest descent method (Lin, 1995), stochastic methods (Shibata and Fukuda, 1994) and genetic algorithms (Vachkov and Fukuda, 1999) could be useful tools for tuning.

Early work on the learning of the RBF network can be applied. For example, Moody and Darken (1989) use a self-organizing technique to find the validity function centre c_i and width σ_i of the receptive fields, and then employ the supervised Adaline or LMS learning rule to identify f_i; Chen et al. (1991) and Wang and Mendel (1992) apply the orthogonal least squares (OLS) algorithm to determine those parameters. However, these approaches couldn't guarantee that the learning is globally optimised, because there is no link made between the local learning and the ultimate criterion function, in which f_i is taken as a scheduling vector only rather than a local model.

Later work in this area involved improving the learning towards a mixed modelling target. Intuitively, an accurate global representation is an essential requirement for modelling, but this doesn't mean that the importance of local modelling could be ignored. Murray-Smith and Johansen (1995) put forward the importance of local learning in LM networks, as follows: "Local learning has a regularizing effect (local models often produce models with higher accuracy, and greater robustness than global learning methods) that can make it favourable compared to global learning in some cases". In the learning and optimising of LM networks, on the one hand, it is necessary to address the significance of global model accuracy; on the other hand, to some extent, the performance of local models in specifically designed local regimes should be guaranteed. While these kinds of learning algorithms can lead to LM networks with arbitrary approximation accuracy, they are computationally expensive.

Brown et al. (1997) present a hybrid-learning scheme towards minimizing the global error in the space of local models. This approach proposes an interpolation function vector $\boldsymbol{\alpha} = [c_1, c_2, \cdots, c_M, \sigma_1, \cdots, \sigma_M]^T$, in which N is the number of local models, as the centers and widths of the normalized Gaussian bell functions. The cost function is rewritten as

$$J(\boldsymbol{\alpha}) = \sum_{k=1}^{M} \left(d_k - \left(\sum_{j=1}^{N} f_j(\boldsymbol{x}) \rho_j (\psi(k), \boldsymbol{\alpha}) \right) \right)^2 \qquad (2.20)$$

where d_k is the scalar plant output, M is the length of the data set and N is the number of local models. Hence the target output is associated with the k^{th} training vector, which includes singular value decomposition (SVD) estimation of the parameters of linear local models and Hessian-based optimisation to identify the nonlinear interpolation functions' parameters. This algorithm is a global optimisation approach in the sense that the parameters of the model are identified using the whole training data set in a single algorithm operation. If sufficient rules and training data are used, the resulting model from these algorithms is guaranteed to converge to the real system.

Johansen and Foss (1995) and Nelles (1997) propose top-down-style construction algorithms from both computational and performance points of view; they optimise the LM network parameters through a repeated loop. In the loop, the algorithm automatically partitions the designed input space into sub-spaces and determines the suitable centres, c_i, and widths, σ_i, for them. In a manner similar to the idea of Moody and Darken (1989), the local learning methods are applied to local model parameter estimation. Based on a similar idea, Vachkov and Fukuda (1999) proposed a structured optimisation scheme to improve learning of fuzzy inference systems.

Another approach which combines local learning and global learning is proposed by Yen et al. (1998), who improve the interpretability of Takagi-Sugeno (TS) fuzzy inference systems by combining local and global learning with weighting functions based on a user's preference. They bring a local MSE criterion function (2.21) to a combined objective function J_C as in (2.22)

$$J_L = \frac{1}{M}\sum_{i=1}^{N}\sum_{k=1}^{M}\left[d(k) - y_i(k)\right]^2 \tag{2.21}$$

where N is the number of local models, M is the length of the data set.

$$J_C = \alpha J_G + \beta J_L \tag{2.22}$$

where α and β are two positive constants satisfying $\alpha + \beta = 1$.

In practice, it is important, for some applications, that the global behaviour of the nonlinear model should accurately represent the global behaviour of the nonlinear system. For example, this is typically the case when the global model is used for nonlinear prediction (Townsend et al., 1998, Townsend and Irwin, 1999) or when the global model is used as an internal model in a controller as in e.g. (Wang, 1993, Brown et al., 1997). On the other hand, it is sometimes required (and often desirable) that the local linear models are accurate approximations to the local linearization. This is the case when the local models are used to design linear local controllers (Narendra and Balakrishan, 1994a, Narendra et al., 1995).

In brief, the learning or optimisation of interpolation schemes is closely related to the objective of controller design. The role of interpolation is to balance the trade-off between local and global approximation accuracy based on the methodology adopted to fulfil the controller design objective.

2.2.5 Structuring Conventional LM Networks

After introducing the regime partition, local models and interpolation schemes, this section brings them together to structure a LM network. A set of simple models defined in each local regime of a nonlinear system is assumed. To each local model, a validity function $\rho_i(\psi(t))$ is associated, which determines the validity of the operating regime for the current operating point defined by $\psi(t)$. As defined in previous sections, $\rho_i(\psi(t))$ is a nonlinear function normalised to satisfy

$$\sum_{i=1}^{N} \rho_i(\psi(t)) = 1 \tag{2.23}$$

in which N is thc number of validity functions. The selection of the scheduling vector is worth mentioning. It is clear that interpolation approaches should reflect the system characteristics at intermediate operating points. Similarly, the choice of scheduling vectors should reflect the instantaneous dynamic nonlinear properties of the system in some appropriate sense. This could be done through a cooperation vector of system states, system input and system output (Murray-Smith and Johansen, 1997). In practice, scheduling vector selection is usually based on the "physics" of the situation, and on the particular characteristics of the model.

2.2.5.1 State Space Representation

Recall the general nonlinear state space system (2.3), we rewrite equation (2.1) in state space format and have

$$\dot{\boldsymbol{x}} = \sum_{i}^{N} \rho_i(\boldsymbol{x},u) f_i(\boldsymbol{x},u) \tag{2.24}$$

where state vector $\boldsymbol{x}$, input u, the model $f_i(:,:)$ is one of N vector local model functions of the state and the input, and is valid in a region defined by the scalar validity function $\boldsymbol{\rho}_i$, which in turn is a function of the above variables.

Typically, the local models are chosen to be of the form as equation (2.7):

$$f_i(\boldsymbol{x},u) = \boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{B}_i u + \boldsymbol{\alpha}_i + h_i(o),$$

resulting in constituent dynamics systems $\sum_i$ given by

$$\textstyle\sum_i: \quad \dot{\boldsymbol{x}} = f_i(\boldsymbol{x},u) = \boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{B}_i u + \boldsymbol{\alpha}_i + h_i(o) \tag{2.25}$$

This results in a nonlinear description of plant dynamics of the form

$$\dot{x} = A(x,u) + B(x,u)u + \alpha(x,u) + e \qquad (2.26)$$

where

$$A(x,u) = \sum_{i=1}^{N} \rho_i(x,u) A_i$$

$$B(x,u) = \sum_{i=1}^{N} \rho_i(x,u) B_i$$

$$\alpha(x,u) = \sum_{i=1}^{N} \rho_i(x,u) \alpha_i$$

$$e = \sum_{i=1}^{N} \rho_i(x,u) h_i(o) \qquad (2.27)$$

Here, $\boldsymbol{\alpha}_i$, $\mathbf{A}_i$ and $\mathbf{B}_i$ are all constants, **e** is the modelling error. Model building consists of covering the operating space of the nonlinear plant with local models. Behaviour along the plant equilibria is typically captured by using models whose equilibria are located inside the region defined by their basis functions, whereas behaviour off equilibria can be properly approximated by using affine models. It is shown in (Johansen et al. 1998a) that the finite set of linearizations about a finite number of points (equilibria and transient points) can be used to accurately approximate dynamic linearization about arbitrary trajectories, using an interpolated multiple model structure with local affine dynamic models.

2.2.5.2 Input-output Representation

In many cases, it is more convenient to use input-output transfer functions rather than state space representation. Recall the general input-output nonlinear system (2.8), if the local models are of ARX form as given in (2.11)

$$f_i(k) = \phi_{ar}^T(k-1)\theta_i + e_i(k) \qquad (2.28)$$

in which,

$$\boldsymbol{\theta}_i = [a_{1i}, \ldots, a_{n_y i}, b_{0i}, \ldots, b_{n_u i}]^T,$$

$$\phi_{ar}(k-1)=[-y(k-1),\ldots,-y(k-n_y),u(k-n_k),\ldots u(k-n_k-n_u)]^T$$

Then LM network (2.1) represents a nonlinear ARX model as follows

$$\hat{y}(k+1)=\phi_{ar}^T(k-1)\sum_{i=1}^{N}\rho_i(\psi(k))\theta_i+e(k) \tag{2.29}$$

The resulting model has an ARX structure at any instantaneous operating time. A major advantage of this model structure is its simplicity and transparency, which leads to more interpretable models; such models are compatible with well-developed control technologies available for linear systems controller design.

2.3 Conventional LM Network Properties and Related Approaches

Having introduced the architecture of LM networks, it is known that LM networks are able to give a close approximation to the modelled nonlinear systems. The basic principle of LM networks is to decompose the entire system operating range into a number of possibly overlapping operating regimes, and in each operating regime a simple local model is applied to describe the local dynamic properties.

This section analyses some properties inherent in LM networks. It starts discussing the number and size of the sub-operating regimes required and the dimensionality problem encountered in LM networks; next, it answers how well LM networks can approximate nonlinear systems; then, it investigates the most commonly used LLM and ALM networks with respect to their approximation ability and potential for controller design. Finally, as an addition, the fuzzy inference system, which is functionally equivalent to LM networks, is concisely introduced; some approaches that are closely related to the LM networks are also discussed.

2.3.1 Fundamental Properties

The core idea of LM networks is to decompose the domain of functions into operating regimes, and to design simpler functions that are adequate approximations to the relationships that are desired for modelling within their respective operating regimes.

Obviously there is a trade-off between the number and size of the operating regimes on the one hand, and the complexities of the local models on the other as shown in Figure 2-4 (Murray-Smith and Johansen, 1997). Provided that the functional relationship approximated is smooth, it should be intuitively clear that approximations based on operating regimes and local models can be made arbitrarily accurate either by making the decomposition into operating regimes sufficiently fine, or by making the local models sufficiently complex.

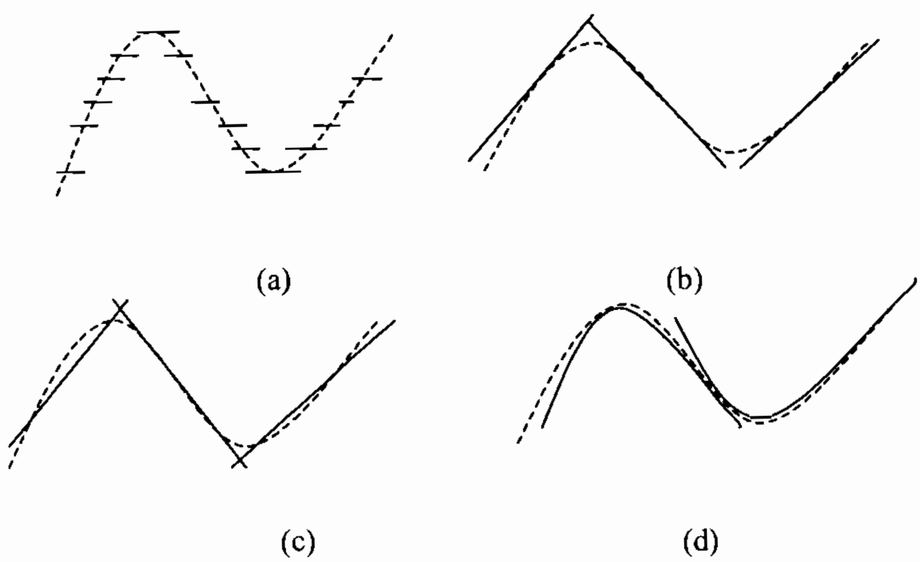


Figure 2-4. Function approximation using local models

The dashed curve is the modelled system, while the solid lines are local approximations. (a) Local constant functions are applied, (b) Local linear functions are applied, (c) Affine linear functions are applied, (d) Local quadratic functions are applied.

It was established that the LM network could uniformly approximate any nonlinear system given a sufficient number of local models N (Johansen and Foss, 1993). The result is based on smoothness conditions on the nonlinear function f, and explicitly formalises the intuitive notion that the operating point scheduling vector $\psi(t)$ should capture the nonlinearities of the system. Moreover, it is shown that any continuous function can be approximated to arbitrary uniform accuracy in the manner discussed using polynomial local models of arbitrarily low order (like linear or constant local approximation). Similar analysis and approximation results are given by (Kosko 1994, Zeng and Singh 1994). These contributions give strong support to the modelling ability of LM networks.

An inherent problem with all function approximation approaches that are based on partitioning of the function's operating regime is the curse of dimensionality: with an increasing number of variables on which the function depends, the number of partitions required (assuming uniform partitioning) will increase exponentially. The use of a priori knowledge about the system is the key to reduce the effects of dimensionality at least to some extent.

In practice, linear or affine models are most widely used as local models. Their advantages are their simplicity and transparency in representation and by their compatibility with well-developed linear control theories and stability analysis techniques. Now, the properties of linear local model (LLM) networks and affine local model (ALM) networks will be investigated, and their advantages and limitations in modelling and control applications will be detailed.

2.3.2 Linear Local Model (LLM)

Conventionally, dynamical modelling of nonlinear systems has been carried out on the basis of linearisation about equilibria. Most often, the ARX/ARMAX structure are selected as local models (Johansen and Foss, 1993) for their simplicity and transparency for controller design. This transparency of representation facilities the incorporation of a priori knowledge, such as known linear regimes and linear models, into the process network (Gawthrop, 1995). Furthermore, the representation is compatible with conventional modelling and control techniques, for example, conventional internal model control (Brown et al., 1997), predictive control (Townsend and Irwin, 1999) and model-based control (Irwin et al., 1998). In particular, it is claimed that the design of a local controller network, once the LLM network has been defined, is relatively straightforward (Townsend et al., 1998, Townsend and Irwin, 1999).

LLM networks clearly inherit many valuable properties for employing linear local models. However, a LM network using strictly local linear models can often result in a poor global representation of the nonlinear plant being approximated due to the side-effects resulted from the normalised basis functions aforementioned. Introducing more linear local models into the network will minimize the error area and provides more potential to reduce the actual steady-state error. But because the network output is bounded by the local models and as such, an infinite number of local models are required to obtain a completely accurate representation (McLoone, 2000).

2.3.3 Affine Local Model (ALM)

As mentioned, the modelling accuracy of LLM networks for nonlinear dynamics is questionable. Dynamic linearization, which means that the linearization is done based on a nominal trajectory, was suggested in (Driankov et al., 1996); however, a drawback is that the control design for the resulting linear time-varying (LTV) system is, in general, a very difficult problem. An alternative is off-equilibrium linearization (Johansen et al., 1998a), which can be seen to depend only on the granularity of the set of points in the off-equilibrium linearization.

The reason for this is that the LTV system resulting from dynamic linearization depends only on the point that the trajectory passes through at a given time. Hence, off-equilibrium linearization leads to an arbitrarily close approximation of the LTV system in terms of a set of linear time invariant (LTI) systems, provided there exists an LTI system close to any point in the nominal trajectory of the LTV system, and the LTI systems are interpolated using a sensible interpolation scheme (Driankov et al., 1996). Mathematically, off-equilibrium linearization leads to local affine models, which have an extra degree of freedom (i.e. an added offset term) to make the local models more flexible, so that they can be shifted upwards or downwards in the operating space (McLoone, 2000).

However, the affine models do not possess the superposition property fundamental to linear systems, due to the inhomogeneous term $\boldsymbol{\alpha}_i$ (equation 2.25). Thus, there is a lack of continuity with established linear theory and methods. Furthermore, the inhomogeneous term can become quite large and significantly influence the solution.

Then $\boldsymbol{a}_i$ tends to dominate, while varying some elements, like $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$, only has minor influence on the local model accuracy (Shorten et al., 1999). Therefore, in general, the inhomogeneous term cannot be simply regarded as a small approximation error or disturbance. A novel velocity-based linearization approach is suggested as an alternative (Leith and Leithead, 1998, 1999). We will discuss this method in a later chapter.

2.3.4 Fuzzy If-then Rules and Inference Systems

Fuzzy inference systems are also known as fuzzy rule based systems, fuzzy models, fuzzy associative memories, or fuzzy controllers. A fuzzy inference system is composed of a set of fuzzy if-then rules, a database containing membership functions of linguistic labels, and an inference mechanism called fuzzy reasoning. The method of fuzzy logic inference modelling was proposed by (Takagi and Sugeno, 1985), and is known as the Takagi-Sugeno (TS) model in fuzzy systems literature. This method has been one of the major topics in both theoretical studies and practical applications of nonlinear modelling and control.

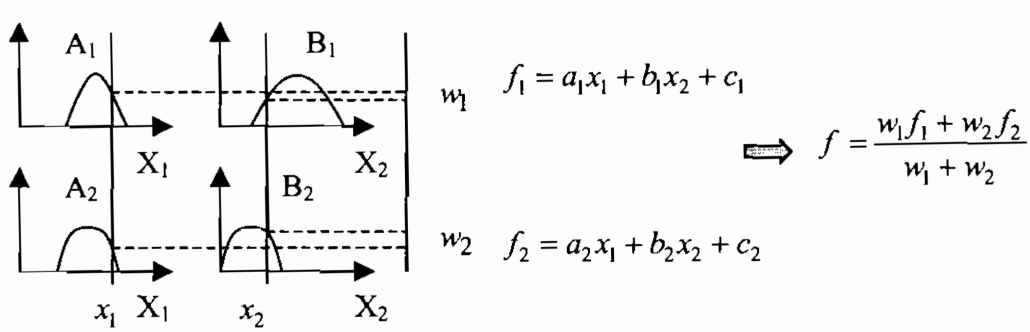


Figure 2-5. Fuzzy inference system example

Suppose there is a rule base consisting of two fuzzy if-then rules of TS type:

Rule 1: If x_1 is A_1 and x_2 is B_1, then $f_1 = a_1x_1 + b_1x_2 + c_1$,

Rule 2: If x_1 is A_2 and x_2 is B_2, then $f_1 = a_2x_1 + b_2x_2 + c_2$, then the fuzzy reasoning mechanism can be illustrated in figure 2-5, where the firing strength (or weight) of the i^{th} rule is obtained as the T-norm (usually the minimum or multiplication operator) of the membership values:

$$w_i = \mu A_i(x_1)\mu B_i(x_2) \text{ or } w_i = \min\{\mu A_i(x_1), \mu B_i(x_2)\}$$

Note, conventionally, that the overall output can be chosen as the weighted average of each rule's output

$$f(x) = \frac{\sum_{i=1}^{R} w_i f_i}{\sum_{i=1}^{R} w_i} \tag{2.30}$$

where R is the number of fuzzy if-then rules.

Generally, a TS model consists of iterative if-then rules that have the form

$$R^i: \text{if } x_1 \text{ is } A_{i1} \text{ and } \dots\ x_k \text{ is } A_{ik},$$

$$\text{then} \quad y_i = p_{i0} + p_{i1}x_1 + \cdots + p_{ik}x_k \ \text{ for } i=1,2,\dots N. \tag{2.31}$$

where N is the number of the rules, x_i are input variables, y_i are local output variables, A_{ij} are the fuzzy sets that are characterized by validity functions $A_{ij}(x_j)$, and p_{ij} are the real valued parameters. The overall output of the model is computed by

$$y = \frac{\sum_{i=1}^{N} \tau_i y_i}{\sum_{i=1}^{N} \tau_i} = \frac{\sum_{i=1}^{N} \tau_i (p_{i0} + p_{i1}x_1 + \cdots + p_{ik}x_k)}{\sum_{i=1}^{N} \tau_i} \tag{2.32}$$

where τ_i is the firing strength of the rule R^i, which is defined as

$$\tau_i = A_{i1}(x_1) A_{i2}(x_2) \cdots A_{ik}(x_k) \tag{2.33}$$

Defining $\rho_i = \frac{\tau_i}{\sum_{i=1}^{N} \tau_i}$ and $f_i = p_{i0} + p_{i1}x_1 + \dots p_{ik}x_k$, then (2.32) can be rewritten as $y = \sum_{i=1}^{N} \rho_i f_i$, which is the same as (2.1) under some minor restrictions:

- The number of local models must be the same.
- The validity functions within each rule are chosen as Gaussian functions with the same variance.
- Both the LM network and TS fuzzy inference systems under consideration use the same method to derive their overall outputs.

The LM network is strongly linked to the fuzzy inference system. Under certain conditions, the TS model of fuzzy inference is functionally equivalent to the LM network. Furthermore, it can be claimed that both models are universal approximations if the validity functions and membership functions are chosen as a scaled version of the Gaussian function. Some contributions have been made by Jang and Sun (1993) and Hunt et al. (1996) in this respect.

Due to the equivalence of these models, it becomes straightforward to apply one model's learning rules to the other, and vice versa. Moreover, from the control point of view, the controller design strategies and stability analysis developed for fuzzy inference systems can be changed to serve the controller design based on LM networks.

A number of other modelling approaches are quite closely related to the LM network. Several authors developed piecewise linear models (without smooth interpolation). These include Skeppstedt et al. (1992) and Billings and Voon (1987). There are many examples of multiple modelling approaches in the statistical literature. These include the Smooth Threshold AR model of Tong and Lim (1980) and the non-parametric state-dependent models of Priestly (1988). These modelling approaches are related with LM networks (2.29) as represented by

$$\hat{y}(k) = \phi^T(k-1)\sum_{i=1}^{n_\psi} \theta_i \rho_i(\psi) + e(k) \tag{2.34}$$

This equation appears as a linear model with parameters, which depend on the interpreting vector (which corresponds to the 'state' in Priestly's terminology). The LM network provides a way of finitely parameterising the state-dependent model.

2.4 Case Study

In order to allow better understanding of the structure and properties of LM networks introduced so far, a case study is carried out. A LM network model was built for the continuous stirred tank reactor (CSTR) simulation, a highly nonlinear simulated process, which has become a standard benchmark in the field of nonlinear system modelling and control studies (Henson and Seborg, 1990). Simulations are carried out in two steps: the first step shows the important influence of interpolation schemes on modelling accuracy; the second step shows the inherent properties and approximation ability of LM networks.

2.4.1 CSTR Introduction

A simplified diagram of the CSTR process is shown in figure 2-6. It is a single input, single output process, where the input is the flow rate of a coolant and the output is the concentration of a product compound.

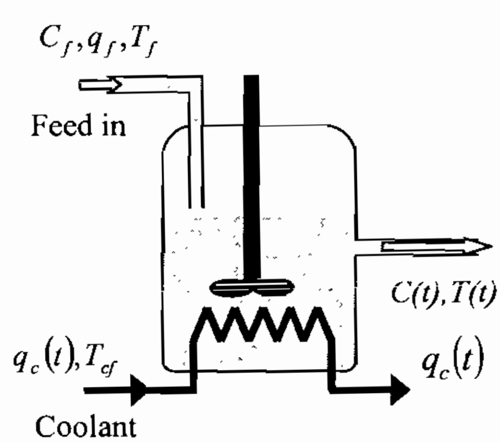


Figure 2-6. CSTR plant model

The reaction that takes place to produce the compound is exothermic which raises the temperature and reduces the reaction rate. The induction of a coolant allows manipulation of the temperature and, hence, control of the concentration. The reaction takes place in container of a fixed volume and the product flowrate, input concentration temperature, and the output flowrate are all assumed constant at their nominal values.

The process model consists of two nonlinear ordinary differential equations (Henson and Seborg, 1990):

$$\dot{T}(t)=\frac{q_f}{V}\left(T_f-T(t)\right)+K_1C(t)\exp\left(-\frac{E}{RT(t)}\right)+K_2q_c(t)\left[1-\exp\left(-\frac{K_3}{q_c(t)}\right)\right]\left(T_{cf}-T(t)\right) \text{ and}$$

$$\dot{C}(t)=\frac{q_f}{V}\left(C_f-C(t)\right)-K_0C(t)\exp\left(-\frac{E}{RT(t)}\right) \qquad (2.34)$$

where $q_c(t)$ is the coolant flow rate, $T(t)$ is the temperature of the solution and $C(t)$ is the output product concentration. The model parameters defined and nominal operating conditions are shown in Table 1 (Henson and Seborg, 1990).

$K_2=0.01 l^{-1}$, constant	K_3=700 l/min, constant
q_f= 100 l/min, product flow rate	V=100 l, container volume
C_f=1 mol/l, input concentration	$E/R=10^4$ K, activation energy
T_f=350 K, input temperature	T_{cf}=350 K, temperature of coolant
$K_0=7.2\times10^{10}\text{min}^{-1}$, constant	$K_1=1.44\times10^{13}$ Kl/min/mol, constant

Table 1. Nominal CSTR Operating Conditions

CSTR is a highly nonlinear system with exponential terms and product terms. The effect may be seen by open-loop step responses over a number of operating values around certain nominal values.

The CSTR process model was simulated at a nominal operating point of $q_c(t)$=100 l/min, $T(t)$= 441.22 K and $C(t)$=0.0882 mol/l. Figure 2-7 is the step response of output product concentration $C(t)$ and temperature $T(t)$ when the coolant flow rate $q_c(t)$ varies from 90 l/min to 110 l/min. The result shows that the output concentration responses vary from the over-damped ($C(t)$<0.1) to the under-damped ($C(t)$>0.1), reflecting the dynamic variation in the CSTR process.

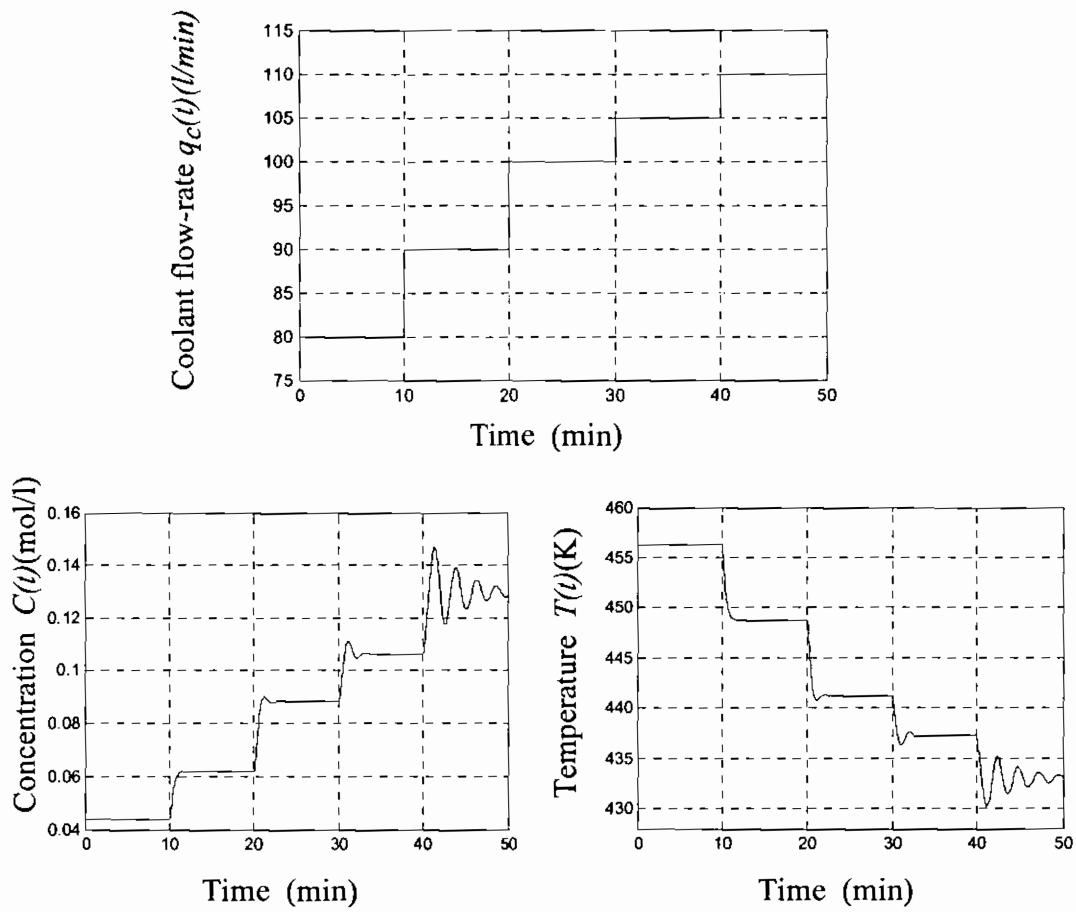


Figure 2-7. Dynamic response of the CSTR plant

In addition, analysis on the linearized CSTR process at nominal operating conditions shows that the stable regime of the CSTR lies in $C(t) \in (0, 0.13566)$ & $q_c(t) \in (0, 110.8)$, which is shown in Figure 2-8.

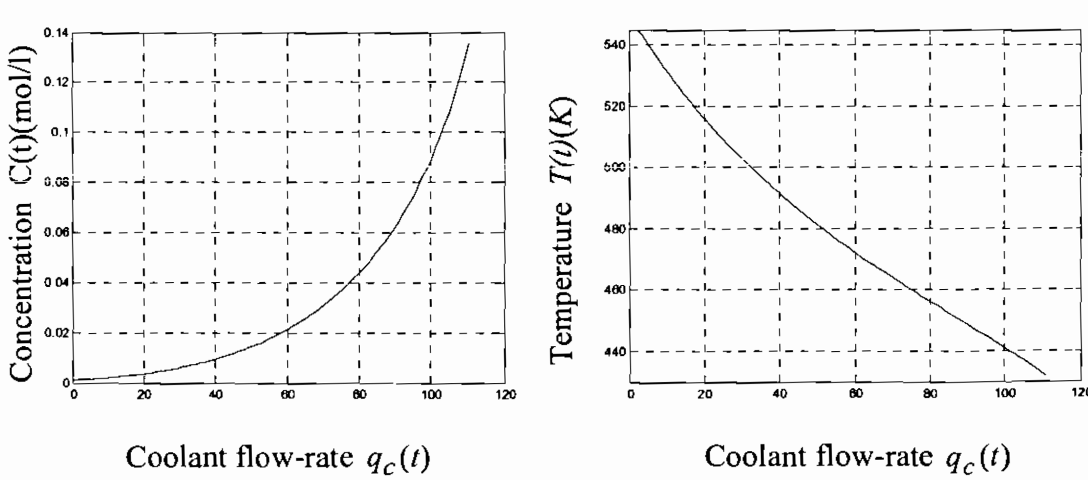


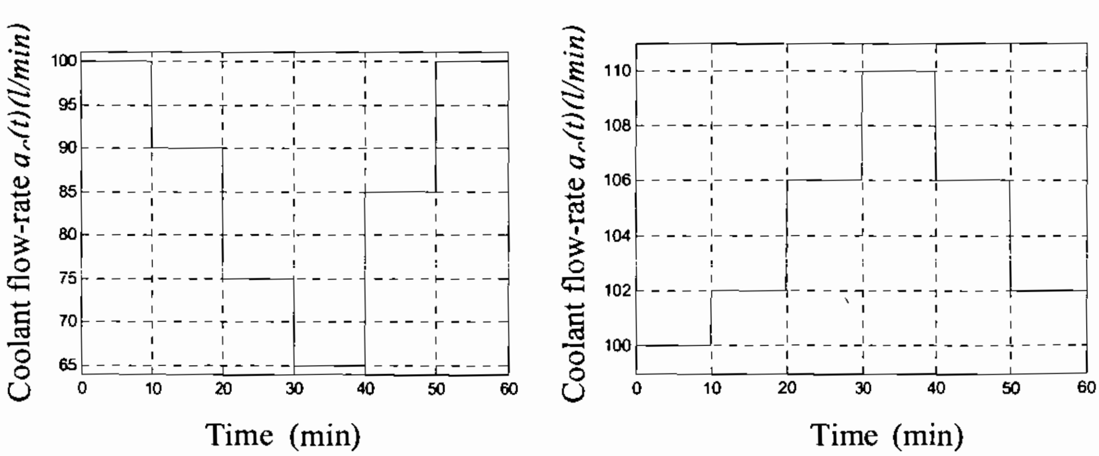
Figure 2-8. Steady-state concentration output from CSTR

For CSTR such complicated nonlinear process, a single linear model is not able to represent the varying dynamic properties. However, a set of models has the potential to describe the dynamic variation in the CSTR process.

2.4.2 Modelling the CSTR Process

The LM network is built to approximate a complex nonlinear system by using a set of local linear (or affine) models. Some factors that have effects on the modelling accuracy are the number and size of partitioned local regimes, the structure and parameters of the local models and the interpolation (or scheduling) schemes used for combining the local models. The difficulty of modelling using LM networks is that it requires deliberate consideration of these mentioned factors, with the hope of using a smaller number of local models to get a good approximation of the modeled process based on the available a priori knowledge of the system.

Two sets of step input signals $q_c(t)$ are chosen, as shown in figure 2-9. One of them covers over-damped operating regime (concentration output *C(t)*<0.1 mol/l) of the CSTR; the other covers the highly under-damped operating regime (concentration output *C(t)*>0.1 mol/l) of the CSTR as shown in Figure 2-6.



(a) over-damped operating regime (b) under-damped operating regime

Figure 2-9. Step changes in coolant flow-rate $q_c(t)$

A desirable LM network, from the point of view of modelling, should be able to give information about the steady state relationship between the input and output signal; meanwhile, it should be able to give the relationship between the input and output signal during transient behaviour. Simulations are carried out to show the ability of LM network in modelling nonlinear systems with certain accuracy from both the static and the dynamic point of view, provided that proper weighting functions have been selected.

First, the full operating range of the CSTR is decomposed to two sub-regimes, in which two local linear models are formulated at following nominal operating points:

$C_o^1 = 0.1298 mol/l, T_o^1 = 432.9487K, q_{co}^1 = 110.0 l/\min$

$C_o^2 = 0.0620 mol/l, T_o^2 = 448.7522K, q_{co}^2 = 90.0 l/\min$

in which (C_o^i, T_o^i, q_{co}^i) denotes the linearization point of the i^{th} local model. The local state space models are

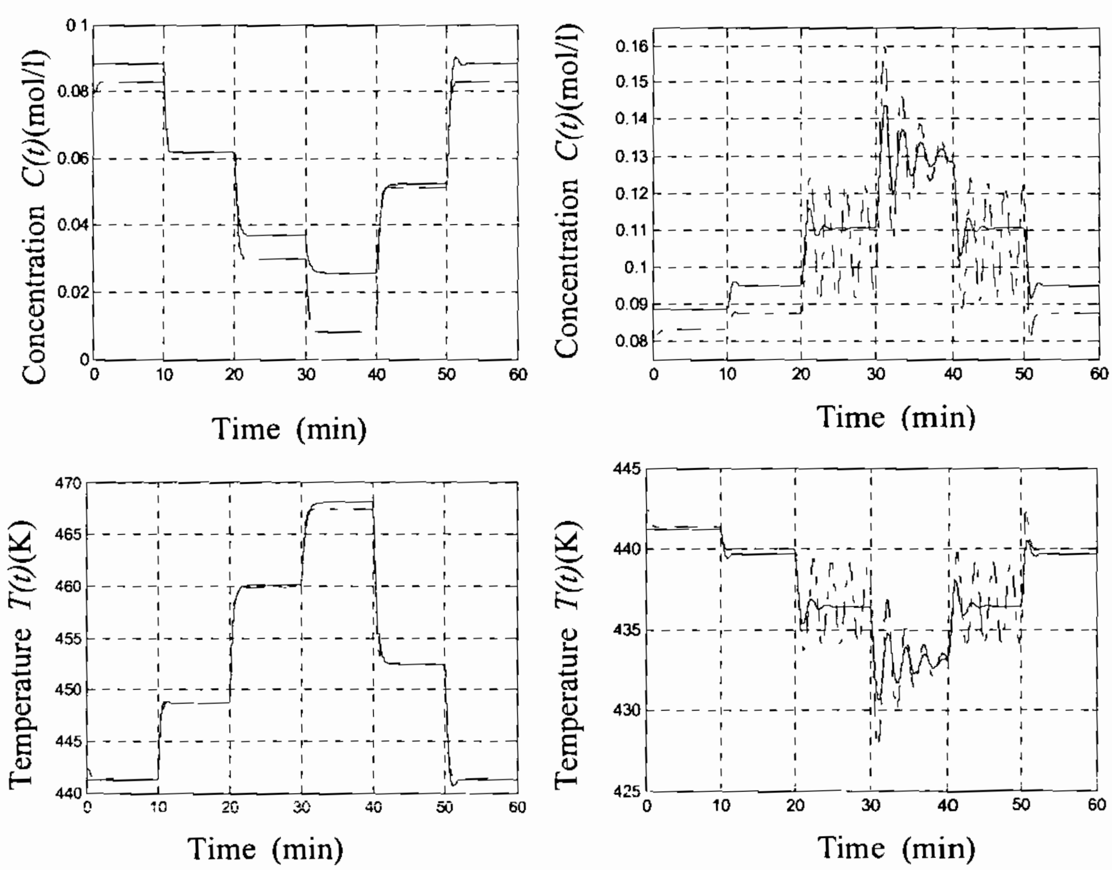
$$A_1 = \begin{bmatrix} 7.1884 & 1.3405e3 \\ -4.6423e-2 & -7.7027 \end{bmatrix},\ B_1 = \begin{bmatrix} -0.8191 \\ 0 \end{bmatrix},\ \alpha_1 = \begin{bmatrix} -3.1961e3 \\ 2.11e1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 7.4161 & 3.0247e3 \\ -4.6577e-2 & -1.6118e1 \end{bmatrix},\ B_2 = \begin{bmatrix} -0.9839 \\ 0 \end{bmatrix},\ \alpha_2 = \begin{bmatrix} -3.4269e3 \\ 2.19e1 \end{bmatrix}$$

in which A_i, B_i, α_i denotes the parameters for the i^{th} local model. The LM network outputs are given in Figure 2-10, which compares the output product concentration *C(t)* and the temperature outputs *T(t)* from the CSTR process with the corresponding outputs from the LM network, when the input signal (coolant flow rate) $q_C(t)$ changes as figure 2-9.

We can see in Figure 2-10 that the LM network with two local models doesn't give accurate performance. In over-damped (Figure 2-10(a)) operating regime, the LM network output is able to follow the slow dynamic changes of CSTR, but there are serious steady state modelling errors existing. In relatively under-damped operating regime, Figure 2-10(b) shows that the LM network could not follow the fast dynamic change in that area. Especially when *C(t)*=0.11, both the concentration output *C(t)* and temperature output *T(t)* are oscillatory.

In addition, it should be noticed that when the operating point is close to the regimes, where the local models are designed, the steady state errors are small. For instance, when *C(t)* is close to 0.062 in Figure 2-10(a) and 0.1298 in Figure 2-10(b). While the steady state errors are getting bigger when the operating point is further away from those regimes. For instance, when *C(t)* is about 0.02 in Figure 2-10(a) and 0.09 in Figure 2-10(b).



(a) Over-damped operating regime (b) Under-damped operating regime

Figure 2-10. Step response Comparison of the CSTR with the LM network with 2 local models

Solid line is from CSTR plant model, the dashed line from the LM networks.

Simulation results given above are when the width vector between the local models is designed as $\sigma = [0.02, 0.03]$. The interpolation scheme is shown as figure 2-11. While these weighting functions are not necessarily optimal, they are more than adequate for the purpose of the main objective of the thesis, i.e. the controller design based on the LM networks. In later chapters, this LM network will be utilized for the local controller (LC) network development with two local controllers.

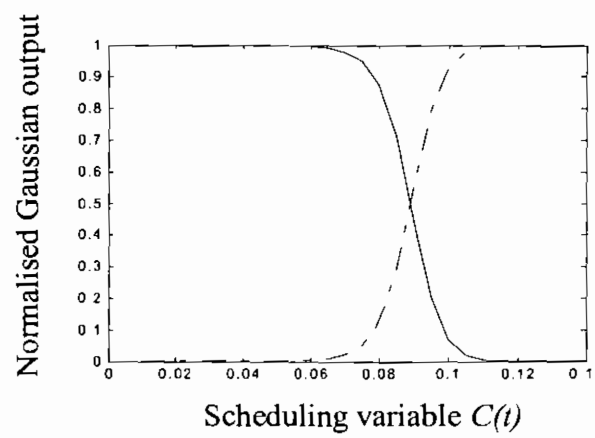


Figure 2-11. Normalised Gaussian based weighting function for the LM network with 2 local models

Increasing the number of local models can limit the steady state errors to an upper bound. This will be shown in the second step of the simulation. Five local models are designed based on the following operating points:

$C_o^1 = 0.1298 mol/l, T_o^1 = 432.95K, q_{co}^1 = 110.0 l/\min$

$C_o^2 = 0.08506 mol/l, T_o^2 = 442K, q_{co}^2 = 98.899 l/\min$

$C_o^3 = 0.05854 mol/l, T_o^3 = 450K, q_{co}^3 = 88.291 l/\min$

$C_o^4 = 0.02946 mol/l, T_o^4 = 465K, q_{co}^4 = 68.788 l/\min$

$C_o^5 = 0.01463 mol/l, T_o^5 = 481K, q_{co}^5 = 50.438 l/\min$

in which (C_o^i, T_o^i, q_{co}^i) denotes the linearization point of the i^{th} local model. The local state space models are

$$A_1 = \begin{bmatrix} 7.1884 & 1.3405e3 \\ -4.6423e-2 & -7.7027 \end{bmatrix},\ B_1 = \begin{bmatrix} -0.8191 \\ 0 \end{bmatrix},\ \alpha_1 = \begin{bmatrix} -3.1961e3 \\ 2.11e1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 7.3783 & 2.1513e3 \\ -4.6832e-2 & -1.1175e1 \end{bmatrix},\ B_2 = \begin{bmatrix} -0.9137 \\ 0 \end{bmatrix},\ \alpha_2 = \begin{bmatrix} -3.3538e3 \\ 2.17e1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 7.4158 & 3.2164e3 \\ -4.6492e-2 & -1.7082e1 \end{bmatrix},\ B_3 = \begin{bmatrix} -0.9968 \\ 0 \end{bmatrix},\ \alpha_3 = \begin{bmatrix} -3.4374e3 \\ 2.19e1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 7.2893 & 6.5871e3 \\ -4.4885e-2 & -3.3936e1 \end{bmatrix},\ B_4 = \begin{bmatrix} -1.1495 \\ 0 \end{bmatrix},\ \alpha_4 = \begin{bmatrix} -3.5046e3 \\ 2.19e1 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} 7.0136 & 1.347e4 \\ -4.259e-2 & -6.8350e1 \end{bmatrix},\ B_5 = \begin{bmatrix} -1.30998 \\ 0 \end{bmatrix},\ \alpha_5 = \begin{bmatrix} -3.5045e3 \\ 2.15e1 \end{bmatrix}$$

in which A_i, B_i, α_i denotes the parameters for the i^{th} local model. The width vector between the local models is designed as σ = [0.03,0.015,0.015,0.012,0.01]. The interpolation scheme based on normalised Gaussian basis function is given in Figure 2-12.

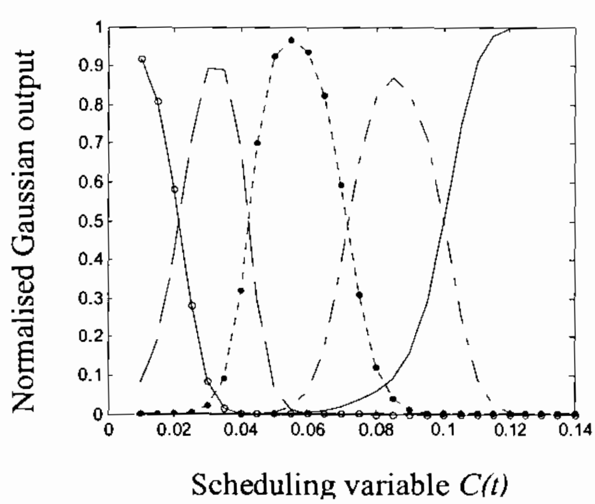
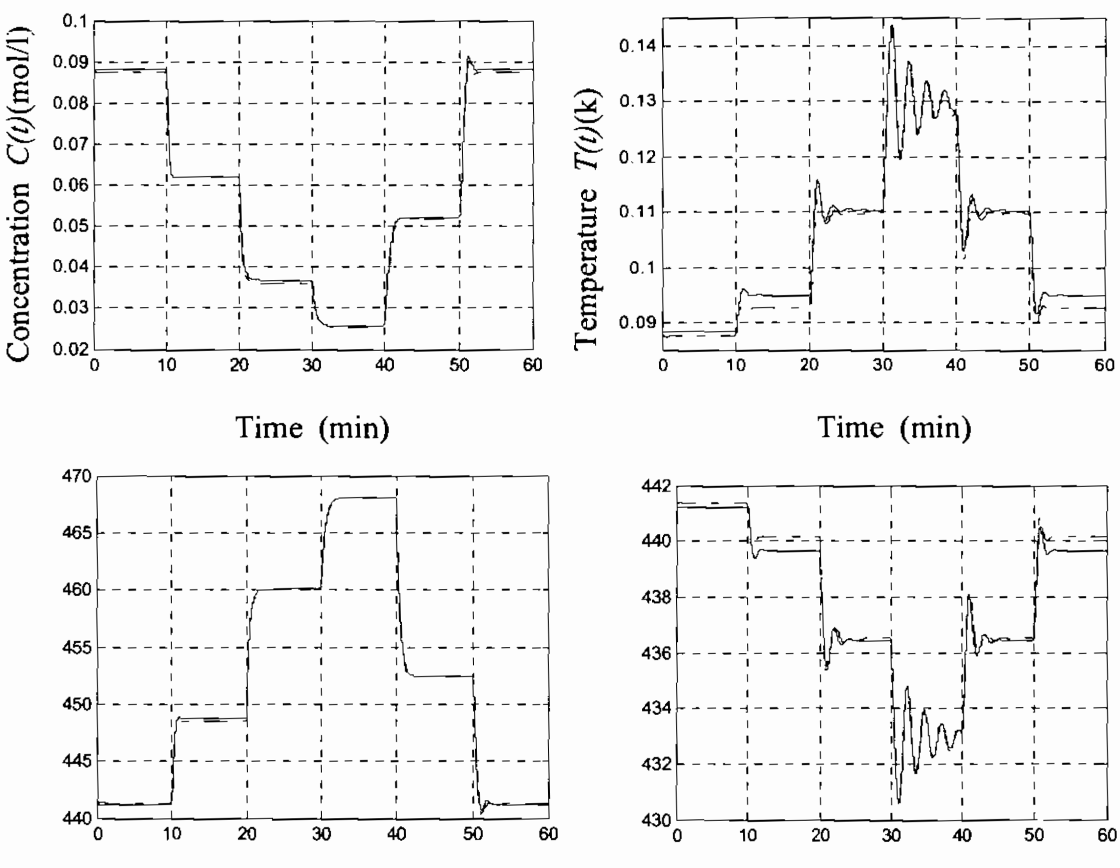


Figure 2-12. Normalised Gaussian basis function for the LM network with 5 local models

Figure 2-13 shows the step response comparison of CSTR with the LM network with five local models. The reasonably improved matching is shown between the LM network outputs and the process CSTR outputs. Steady state errors are decreased significantly all over the operating space comparing with Figure (2-10), no matter in over-damped operating regime (Figure 2-13(a)) or in under-damped operating regime

(Figure 2-13(b)). It is worthwhile noting that, when the coolant flow rate $q_c(t)$=106 l/min, $C(t)$=0.11, the CSTR output is highly under-damped; however, the LM network output still matches the CSTR output with high accuracy, indicating the better modelling accuracy than the LM network with two local models.



(a) Over-damped operating regime (b) Under-damped operating regime

Figure 2-13. Step response comparison of the CSTR with the LM network with 5 local models

Solid line is from CSTR plant model; dashed line is from the LM network.

In summary, LM networks are able to model nonlinear systems with certain accuracy. The steady state error between process and model can be limited to an certain bound if a proper tradeoff can be reached among the modelling issues, such as the number of local models, the structure of local models and the interpolation techniques involved.

2.5 Concluding Remarks

This chapter considered the issues of structuring the LM networks and investigated the properties of utilizing LM networks for modelling nonlinear systems. Theoretically, LM networks are capable of approximating nonlinear systems with certain accuracy by employing a finite number of linear local models. However, in practice, there are normally modelling errors involved in the approach, which, in part, result from the normalization of validity functions.

Generally, in the operating points where the local model is defined through linearization, accurate approximation can be made without much difficulty; however, this is not true for the operating regime far away from these points. The further away the operation regime is from these special operating points, the bigger the steady state error. Increasing the number of local models and/or optimizing the interpolation schemes may improve the modelling accuracy and be able to decrease the steady state errors to an upper bound globally to satisfy the desired modelling requirement.

From the modelling point of view, the LM network can incorporate a priori knowledge of a physical system into the framework of the model and have the framework optimised through the learning of the LM network. Such kind of modelling structure carries on the advantages of linear models for controller design, i.e. the well-developed control techniques for linear systems can be extended to nonlinear systems through the LM network structure without much difficulty.

In the following chapters, we will investigate the controller design and system analysis techniques for nonlinear systems based on the LM networks, by employing the control methods and theory that have been well developed for linear systems.

Chapter 3

Local Controller Networks

3.1 Introduction

LM networks can be referred to as universal approximators due to their capability for approximating a given system with arbitrary accuracy. The general idea of LM networks is to decompose the operating state space into small local regions and to approximate the system in every region by a simple (usually linear or affine) model. The overall LM network is considered to be a combination of interconnected systems with simple local models. This chapter describes nonlinear controller design based on LM networks.

From the modeling point of view, LM networks provide an excellent basis for developing good data-driven identification methods that require little a priori knowledge of the system under investigation. The model structure (partitioning the full operating range and employing local model structures) and local model properties can, in some applications, be relatively easily related to the physics of the system. This simplifies model development and validation. Meanwhile, the relatively complex interpolation technique allows the number of local models to be quite small in many applications.

From a control engineering perspective, the use of local linear (or affine) models bridges the gap between nonlinear control and conventional control. Many existing tools and theories in linear systems can be applied to LM networks to complete the controller design. This is one of the significant advantages of applying LM networks for control purposes compared to the more usual approach based on a complete nonlinear plant model, for example neural networks, which inevitably suffers from computation penalty due to the requirement for nonlinear optimization to minimize the cost function. LM networks have been successfully used in such control schemes as internal model control (Brown et al. 1997), predictive control (Townsend et al., 1998, Townsend and Irwin 1999, Gao et al., 2002b), and PID controllers (Gao et al., 2002a).

Linear system theory is based upon linearization of the underlying nonlinear dynamics in some partitioned regions around some nominal operating points. Traditionally, these operating points correspond to physical equilibria of the plant. However, if the operating point is moved away from the nominal operating point, the local controller is less effective, or even detrimental to the system. It is intuitive that a controller designed around a specific operating point may not be able to accommodate large variations in process dynamics and offer satisfactory tracking control without some performance degradation or even instability.

One solution is the use of multiple models, which provides a convenient framework for obtaining both stability and improved performance simultaneously. Gain scheduling is perhaps one of the most popular applications of multiple models (Shamma and Athans, 1990). Gain scheduling conventionally constructs a non-linear controller, with certain required dynamic properties, by combining, in some sense, the members of an appropriate family of linear time-invariant controllers. The transition between the family members is governed normally by an auxiliary scheduling variable; a traditional rule of thumb to ensure stability is that the scheduling variables vary slowly with respect to the states and are able to capture parameter variations of the controlled nonlinear system (Shamma et al. 1991).

Another closely related approach is piecewise (i.e. no interpolation between local models) multiple model adaptive control, which was suggested by Middleton et al.

(1988), and further developed by Narendra and Balakrishnan (1994a, 1997). This strategy employs different classes of switching and tuning schemes, to 'combine' the fixed and adaptive models, to improve the transient performance. Further investigation is given by Narendra and Cheng (2000) on the overall global system stability and on the convergence of tracking errors to zero. This investigation shows that an arbitrary switching scheme yields a globally stable system, provided that the interval between successive switches has an arbitrarily small but non-zero bound. These results strongly support the application of this approach in industry. Compared with the gain scheduling approach, where the local controllers are scheduled in open loop, the piecewise multiple model adaptive control is scheduled based on the modelling performance of the local models.

As mentioned, the objective of this chapter is to present the formulation of the controller designs based on LM networks. The theoretical principle for the design of the controller using LM networks are based on well-established results for linear systems. Using the implicit function theorem, the properties of the linearized system around the equilibrium states can be extended to the nonlinear domain, assuming a LM network has been created with arbitrary accuracy for a given nonlinear system.

Generally, there are two approaches to complete a controller based on LM networks. One approach is to structure the local controller (LC) network based on locally valid models, then the global controller is formulated by the combination of the local controllers; the other approach is to design a global LPV (Linear Parameter Varying) controller based on a global nonlinear model resulting from the LM network.

Controller design based on the locally valid model could be called a specialized interpolation technique for the gain scheduling concept (Hunt and Johansen, 1997, Johansen et al., 1998a, 1998b, Korba and Frank, 2000, Sharma et al., 2002). This approach allows the design of a locally valid controller for each local model, and the global controller output is given by the combination of local controller outputs.

In contrast, the global design method uses the dynamic LM network global model for the controller design (Brown et al., 1997, Brown and Irwin, 1999). This global model can be considered as a linear parameter varying (LPV) system, which is an approximation of the instant linearization of the controlled nonlinear system, so LPV theory can be employed for the controller design and for the stability analysis (Rubensson and Lennartson, 2000, Rong, 2002).

The contents of this chapter are as follows. Firstly, in section 3.2, the development of the gain scheduled LC network is proposed by employing a model based predictive control strategy. In section 3.3, Case studies are given to show how the linear control techniques can be extended to the control of nonlinear systems and to bring out some new issues related to the blending techniques on the control of a highly nonlinear simulated process. Section 3.4 introduces the LPV controller design strategy based on a global LM network model. In section 3.5, the concluding remarks summarize the main issues discussed in this chapter.

3.2 Gain-scheduled Local Controller Networks

3.2.1 Gain Scheduling

Gain scheduling is a well-established engineering practice to design controllers for nonlinear plants by linearizing the nonlinear dynamics at some fixed representative operating points (Shamma and Athans, 1990, Rugh, 1991). The synthesis of a nonlinear control system is made by objectively combining a family of linear controllers. A typical gain-scheduled design procedure for nonlinear plants after selecting a number of operating points is as follows:

1. Linearize the plant about a finite number of representative nominal operating points, which cover the whole range of the nonlinear system's dynamics.
2. Design a linear controller at each of these nominal operating points.
3. Interpolate the parameters of the linear controllers, designed at step 2, to achieve adequate performance of the closed loop system at all points where the plant is

expected to operate.

In the case of gain scheduling on a reference state trajectory, these operating points are time-frozen values of the state vector that belong to the reference state trajectory. At each of the nominal operating points, a local linear approximation of the original nonlinear plant is constructed and a local linear controller is designed for each local linearized plant. For intermediate operating points, the local controllers are interpolated or scheduled, thus resulting in a global gain scheduling control or gain scheduler.

The main advantage of the gain scheduling approach is that linear design methods are applied to the linearized system at each operating condition, and the wealth of linear control methods, performance measures, design intuition, and computational tools can be brought to bear on control design for general nonlinear systems. For example, pole placement, quadratic performance indices, output feedback control techniques and model predictive control can be used in specifying performance. Meanwhile, model-based methods, such as adaptive and robust design for linear systems can be applied to counter uncertainty in system parameters, which will benefit the final gain-scheduled design. Moreover, a gain-scheduled control system has the potential to respond rapidly to changing operating conditions.

However, these significant advantages are partly offset by some difficulties, such as the selection of appropriate scheduling variable and the selection of the scheduling procedure. General characterizations and prescriptions have not been particular useful, except for the rules of thumb, such as "schedule on a slow variable". In this case, most of the theory developed for scheduling is likely to simply verify the intuition obtained from an understanding of a physical system (Rugh, 1991); analyzing the global control performance and guaranteeing the global stability in the presence of rapid variations is thus difficult.

An alternative gain-scheduling framework should be expected to give guarantees of expected global performance, stability and robustness of the closed-loop system in the presence of rapidly changing operating conditions. The gain scheduled LC network shows some progress towards this objective.

The LC network has the structure of a two-level control system. The first level is to provide a basic feedback control and the second level is that the interpolation schemes provide supervisory function. By using a LM network, these two kinds of knowledge can be formulated into a unified mathematical framework. Moreover, the LM network can incorporatc a priori knowledge of a physical system into the framework of the model and have the framework optimised through the learning of the LM network; this improves the conventional scheduling technique, which is based on intuition of the physical system. This framework brings more flexibility for gain scheduling and provides the possibility of developing a systematic analysis and design method for complex nonlinear control.

Some research work has been done by Hunt and Johansen (1997) and Johansen et al (1998a, 1998b) under this heading. One obvious potential benefit from this approach is the improved transient performance over the full operating range when using an affine LM network. Other approaches, like the fuzzy inference system, which is functionally equivalent to LM networks, also can be applied for formulating gain-scheduled control (Driankov et al. 1996, Farinwata et al., 2000, Bergsten et al., 2000, Lin, 2002).

The following sections show how to properly utilize the well-developed linear control theories and analysis techniques for gain-scheduled LC networks. A general framework of LC networks is given by introducing the generalized predictive controller into the framework of the local controller network.

3.2.2 LM Networks as Gain-scheduled Systems

Consider a single input single output (SISO) LM network model $\hat{y}$ that is described by equation (2.1). We rewrite it as follows for convenience:

$$\hat{y}(t) = \sum_{i=1}^{N} \rho_i(\psi(t))\hat{y}_i(x) \tag{3.1}$$

where $i = [1, N]$, $\hat{y}_i$ is the local model output, $\hat{y}$ is the global LM network output and ρ_i

are the normalized weights as introduced in chapter two. The weight vector $[\rho_1,\cdots,\rho_N]$ is a function of the scheduling variables $\psi(t)$, which, in the framework of LM networks, can be a collection of system inputs or outputs, or states. The selection of scheduling variables $\psi(t)$ should explicitly address the possibility of rapid system variations.

Overall, the LM network model is nonlinear, with parameters that are scheduled based on the weight vector $\rho=[\rho_1,\cdots,\rho_N]$, in which ρ_i is a nonlinear function. In a broad sense, the LM network can be considered as a gain-scheduled system with the firing levels as scheduling variables. If the scheduling vector resides in only one i^{th} region, then the LM network coincides with the i^{th} linear local model. Because of the partitioning of the operating range, LM networks may belong to more than one i^{th} regime and in this case the output of the LM network is formed by a combination of the subsystems that are associated with these regimes.

LC networks can be built in a straightforward manner, based on the LM network structure. The basic idea is to adaptively blend locally valid controllers at different operating regions of the process smoothly and effectively through the interpolation system, which results from the LM network. Based on N locally valid models, the nonlinear controller design problem becomes the problem of designing N local controllers, using available linear control methods and theories, to achieve the desired performance in each selected operating regime of the nonlinear system. Then the local controller (LC) network combine these locally designed controllers to formulate a globally valid controller which covers the full operating range.

In general, the global control signal is defined by

$$u(t)=\sum_{i=1}^{N}C_i\left(\phi^c(t)\rho_i(\psi(t))\right) \tag{3.2}$$

where C_i denotes the local controller for each local model $\hat{y}_i$. The N local controllers thus obtained are blended using the same validity function ρ_i as the one used in the LM network. The controller information vector ϕ^c could consist of past control inputs, current and past plant outputs, and the current and past values of the reference signal r, depending on the local controller technology applied. Figure 3-1 shows a LC network with an interpolation system.

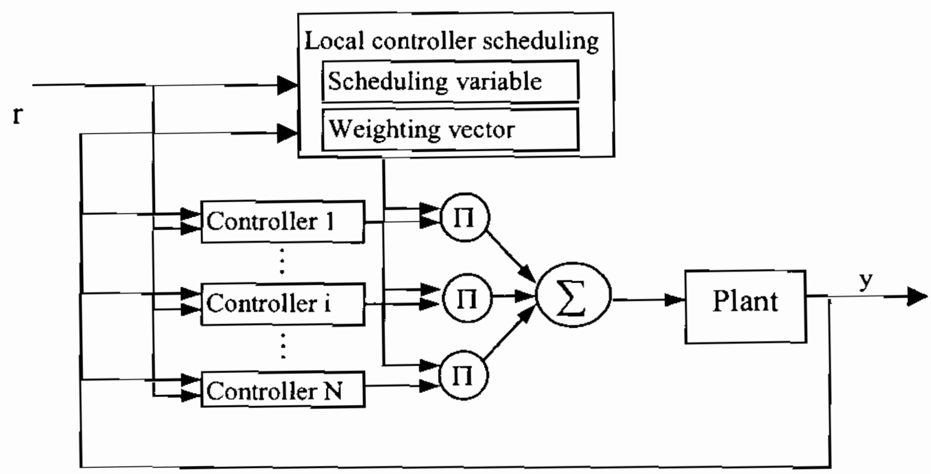


Figure 3-1. Local controller network scheme

One more issue that is worth mentioning is the link between the LC network and fuzzy control. As a control concept, the LC network operating regimes can be viewed as fuzzy sets, and the weighting functions ρ_i interpolating the parameters are functionally equivalent to membership function of Takagi and Sugeno (TS) fuzzy inference systems. It is evident that the LC network approach has a very close resemblance to some TS fuzzy model-based-control schemes such as those described in (Takagi and Sugeno, 1985, Tanaka and Sugeno, 1992, Driankov et al., 1996, Filev, 1996). So the theories and analysis techniques on fuzzy controller design can be used for LC network design and vice versa.

3.2.3 Local Controller Design

The LM network model naturally defines the mechanism for interpolation between the linear local models. In the LM network based gain-scheduled approach, combinations of linear subsystems that are associated with a given region of the operating range are considered. Subsequently, combinations of linear controllers that are assigned to the linear subsystems of a particular overlapped region are considered.

In general, any linear control design method can be applied to design local controllers in a gain scheduled LC network framework. In chapter 2, it was pointed out that there are normally modeling errors existing in the LM network model, although the modeling error can be made arbitrarily small by taking a large number of operating regimes that cover the full operating range with sufficient accuracy. In the development of local controller design, both modeling error and disturbances should be considered as important issues to enable desired controller performance (Hunt and Johansen, 1997).

An affine local model has better modelling performance, for it includes an offset (inhomogeneous) term α_i, which brings extra freedom for modelling. In terms of control, the LC network approach goes a step further by incorporating affine local models for operating regimes which do not necessarily contain any physical equilibria, as the conventional gain-scheduled approach requires.

In this section of the chapter, controller design based on an affine local model takes the extra offset term α_i in the affine local model as a known constant disturbance. This approach improves the system transient response but does not influence its stability when α_i is limited to some degree. However, when α_i is beyound the limit, the stability properties of the affine local model networks will be very different. This phenomenon asks for attention to the affine terms when designing the controller.

The overall global closed-loop system stability depends on the local closed-loop systems and the interpolation techniques between these local systems. Moreover, for some systems which change not 'slowly enough', the offset term should be also critically considered in the controller design. An alternative method that overcomes this problem of using affine models is proposed by Leith and Leithead (1998, 1999, 2001). This method is termed as a velocity-based approach, which will be discussed in chapter five.

Consider a single linear model of the form of equation (2.14)

$$A(q^{-1})y(k) = q^{-k}B(q^{-1})u(k) + e(k) \tag{3.3}$$

where $e(k)$ includes modeling error and disturbance contributions. One possibility for dealing with the error $e(k)$, is to use feedforward control, if knowledge of $A(q^{-1})$, $B(q^{-1})$ and $e(k)$ is available. When an affine local model is applied in the LM network structure, $e(k)$ is partially determined. The determined part is termed as α_0, the same as α_0 in equation (2.7), and $e(k)$ can be rewritten as

$$e(k) = \alpha_0 + \varepsilon(k) \tag{3.4}$$

where $\varepsilon(k)$ denotes unmodelled error, which could result from modelling uncertainties and disturbances existing in any realistic problem. Hunt and Johansen (1997) proposed a feedforward filter to eliminate the influence of the offset term in the formulation of the LC network based on pole-placement feedback control. An alternative is to employ integral action, which removes the requirement for knowledge of $e(k)$. Employing an integrator may be more attractive, since normally the knowledge available of $e(k)$ based on LM network model is less than the knowledge available of $A(q^{-1})$ and $B(q^{-1})$. Moreover, in many practical applications, the system variation is reasonably slow so that the modelling error could be taken as constant.

Conventionally, the PID (proportional-integral-derivative) controller is the most widely deployed in industry. This algorithm is well understood, and relatively easy to tune and has capability to handle certain nonlinearities. Advanced control approaches include adaptive control, internal model control and model predictive control. In recent years, predictive control methods have become a very important area of research. Their application in industry, especially in chemical process control (Morari and Lee, 1999, Qin and Badgwell, 1997) has systematically increased during the last three decades. The fundamental predictive control methods include Model Algorithmic control (Richalet et al., 1978,), Dynamic Matrix control (Cutler and Ramaker, 1980), Extended Prediction Self-adaptive Control (De Keyser and Van Cauwenberghe, 1985), Extended Horizon Adaptive Control (Ydstie, 1984) and Generalized Predictive Control (Clarke et al., 1987a, 1987b, Clarke and Mohtadi, 1989). All of these methods can be generalized in a unified predictive control approach proposed by Soeterboek (1992).

The principle of predictive control is based on the prediction of the process output. The prediction is made implicitly or explicitly according to the model of the process to be controlled and represents the first step in the predictive control algorithm. In the next step, the control signal is selected which brings the predicted process output signal back to the reference signal, in a manner that minimizes the area between the reference and the output signal. Controllers based on those methods are capable of controlling "difficult" processes such as processes with long time delay and non-minimum phase and unstable processes, and can include constraints into the controller design.

However, these significant advantages are partly offset by some difficulties when the predictive control is designed based on nonlinear models. For example, the requirement for nonlinear optimization to minimize the cost function for control is a problem, because it normally involves intensive computation. To overcome the drawback, a LM network can be incorporated into a model predictive control structure to formulate a nonlinear MPC (Kouvaritakis and Cannon, 2001).

In this section, particularly, the generalized predictive control (GPC) approach (Clarke, 1987 a, 1987 b), which has integral action, is introduced for local controller design. Considering regulation about a particular operating point, a non-linear plant generally allows a locally linearized CARIMA model (Controlled Auto-Regressive and Integrated Moving Average model) representation of the disturbances and modeling errors:

$$A\left(q^{-1}\right)y(k)=B\left(q^{-1}\right)u(k-1)+C\left(q^{-1}\right)\xi(k)/\Delta \qquad (3.5)$$

where A and B are polynomials in the backward shift operator q^{-1} as detailed in equations (3.6) and (3.7), Δ is the differencing operator $1\text{-}q^{-1}$, which brings the integrator action and eliminates the steady state errors to zero.

$$A\left(q^{-1}\right)=1+a_1q^{-1}+\cdots+a_{n_y}q^{-n_y} \tag{3.6}$$

$$B\left(q^{-1}\right)=b_0+b_1q^{-1}+\cdots+b_{n_u}q^{-n_u} \tag{3.7}$$

If the plant has a non-zero dead-time, the leading elements of the polynomial $B\left(q^{-1}\right)$ are zero. In (3.5), $u(k)$ is the control input, $y(k)$ is the measured variable or output, and $\xi(k)$ is an uncorrelated random sequence. For simplicity, $C\left(q^{-1}\right)$ is chosen to be 1.

To derive a j-step ahead predictor of $y(k+j)$ based on (3.5) consider the identity:

$$1=E_j\left(q^{-1}\right)A\Delta+q^{-j}F_j\left(q^{-1}\right) \tag{3.8}$$

where E_j and F_j are polynomials uniquely defined, given $A\left(q^{-1}\right)$ and the prediction interval j. Clarke et al. (1987) presented a simple scheme to use recursion of the Diophantine equation so that the polynomials E_{j+1} and F_{j+1} are obtained, given values of E_j and F_j. If (3.5) is multiplied by $E_j\Delta q^j$, we have:

$$E_jA\Delta y(k+j)=E_jB\Delta u(k+j-1)+E_j\xi(k+j) \tag{3.9}$$

and substituting for $E_jA\Delta$ from (3.8) gives

$$y(k+j)=E_jB\Delta u(k+j-1)+F_jy(k)+E_j\xi(k+j) \tag{3.10}$$

As $E_j\left(q^{-1}\right)$ is of degree $j\text{-}1$, the noise components are all in the future, so that the optimal predictor, given measured output data up to k, and any given $u(k+i)$ for $i>1$, is clearly:

$$\hat{y}(k+j\mid k)=G_j\Delta u(k+j-1)+F_jy(k) \tag{3.11}$$

where $G_j(q^{-1}) = E_j B$ and $\hat{y}(k + j \mid k)$ denotes the predicted output.

Suppose a future set-point or reference sequence [$w(k+j)$; j=1,2,...] is available. The objective then of the predictive control law is to drive future plant outputs $\hat{y}(k + j)$ close to $w(k+j)$ in some sense. A cost function of the form in equation (3.12) can be described:

$$J(N_1, N_2) = E\left\{ \sum_{j=H_m}^{H_p} [\hat{y}(k+j) - w(k+j)]^2 + \sum_{j=1}^{Hc} \lambda(j)[\Delta u(k+j-1)]^2 \right\} \quad (3.12)$$

where H_P denotes the prediction horizon; H_m denotes the minimum-cost horizon respectively, H_C denotes the control horizon, and $\lambda(j)$ denotes a control weighting sequence.

Recalling equation (3.10), we have the future outputs:

$$\hat{y}(k+1) = G_1 \Delta u(k) + F_1 y(k) + E_1 \xi(k+1)$$

$$\hat{y}(k+2) = G_2 \Delta u(k+1) + F_2 y(k) + E_2 \xi(k+2)$$

$$\vdots$$

$$\hat{y}(k+N) = G_N \Delta u(k+N-1) + F_N y(k) + E_N \xi(k+N) \quad (3.13)$$

in which $\hat{y}(k + j)$ consists of three parts: one depends on future control actions, one depends on past known systems output, and one depends on future noise signals. The assumption that the controls are performed in open loop ignores the future noise sequence $\xi(k + j)$ in calculating the predictions. Let $f(k + j)$ be that component of $\hat{y}(k + j)$ composed of signals which are known at time k, so that

$$f(k+1) = [G_1(q^{-1}) - g_{10}]\Delta u(k) + F_1 y(k)$$

$$f(k+2) = q[G_2(q^{-1}) - q^{-1} g_{21} - g_{20}]\Delta u(k) + F_2 y(k)$$

$$\vdots$$

$$f(k+N)=q^{N-1}[G_N(q^{-1})-q^{-1}g_{N,N-1}-\cdots-g_{N,0}]\Delta u(k)+F_N y(k),$$

where $G_i(q^{-1})=g_{i0}+g_{i1}q^{-1}+\cdots$

Then the equation above can be written in the key vector form:

$$\hat{y}=G\tilde{u}+f \qquad (3.14),$$

where all the vectors are $N\times 1$:

$$\hat{\mathbf{y}}=[\hat{y}(k+1),\hat{y}(k+2),\cdots,\hat{y}(k+N)]^T$$

$$\tilde{\mathbf{u}}=[\Delta u(k),\Delta u(k+1),\cdots,\Delta u(k+N-1)]^T$$

$$\mathbf{f}=[f(k+1),f(k+2),\cdots,f(k+N)]^T$$

The matrix **G** is a lower triangular matrix of dimension $N\times N$:

$$\mathbf{G}=\begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \cdots & \cdots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_0 \end{bmatrix}$$

From the definition of the vectors above, and with

$$w=[w(k+1),w(k+2),\cdots,w(k+N)]^T,$$

equation (3.12) can be written as:

$$J=\{(G\tilde{u}+f-w)^T(G\tilde{u}+f-w)+\lambda\tilde{u}^T\tilde{u}\} \qquad (3.15)$$

The minimization of J (assuming no constraints on future controls) results in the projected control-increment vector:

$$\tilde{\boldsymbol{u}} = \left(\boldsymbol{G}^T\boldsymbol{G} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{G}^T(\boldsymbol{w} - \boldsymbol{f})$$

Note that the first element of $\tilde{\boldsymbol{u}}$ is $\Delta u(k)$, so that the current control $u(k)$ is given by:

$$u(k) = u(k-1) + \bar{g}^T(w - f) \qquad (3.16),$$

where $\bar{g}^T$ is the first row of $\left(\boldsymbol{G}^T\boldsymbol{G} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{G}^T$. Hence, the control includes integral action, which provides zero static offsets provided that a constant set point $w(k+j) = w$ exists. Operational constraints of system inputs and states can be incorporated into the optimization procedure in the usual manner (Soeterboek, 1992). For each of the local models, a local GPC can be constructed, and a predictive control action is obtained using equation (3.16). Then the global control action is formulated by combining all the local controllers through the interpolation systems, as described in equation (3.2).

To analyze the stability or robustness of a GPC system, it is required to write the predictive control law in polynomial form. If $\xi = 0$, the predictive control law can be written as (Soetebrboek, 1992, Clarke, 1987a, 1987b):

$$Ru(k) = -Sy(k) + Tw(k + H_p) \qquad (3.17)$$

Here, H_p is the prediction horizon; S, R and T are polynomials in the backward shift operator:

$$S(q^{-1}) = s_0 + s_1q^{-1} + \cdots + s_{ns}q^{-ns}$$

$$R(q^{-1}) = r_0 + r_1q^{-1} + \cdots + r_{nr}q^{-nr}$$

$$T(q^{-1}) = t_0 + t_1q^{-1} + \cdots + t_{nt}q^{-nt}$$

The control law above is the one used in the pole-placement controller design method. In this case, $w(k + H_p)$ is equal to the set point. Figure 3-2 shows the control law (3.17) in a block diagram.

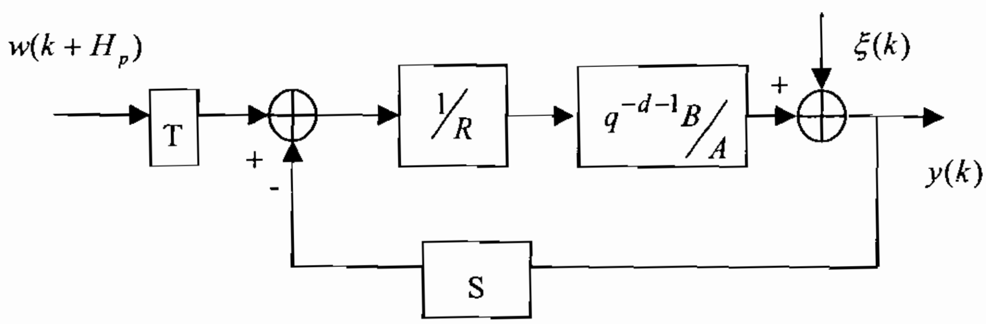


Figure 3-2. The pole placement closed-loop system

From Figure 3-2, the following transfer functions can be calculated:

$$y(k)=\frac{q^{-d-1}BTw(k+H_p)+AR\xi(k)}{AR+q^{-d-1}BS} \tag{3.18}$$

$$u(k)=\frac{ATw(k+H_p)+AS\xi(k)}{AR+q^{-d-1}BS} \tag{3.19}$$

Clearly, to make sure that the predictive system is stable, $AR+q^{-d-1}BS$ should have all its roots inside the unit circle.

3.2.4 Structure of LC Networks

Thus far, the predictive controller is designed only for local linear systems. To structure a LC network, a finite number of N operating regimes ψ_i should be considered. Assume that in each nominal operating regime, a linear local model characterizes the system

$$A_i(q^{-1})y(k)=q^{-k}B_i(q^{-1})u(k)+e(k) \tag{3.20}$$

For each local model, a corresponding local controller is designed (equation (3.16)) with control parameters ($\bar{g}_i^T$, f_i). For each nominal operating regime ψ_i , the system is characterized by a local model to a degree given by validity function ρ_i . The overall global controller is then defined naturally using the validity functions ρ_i as smooth interpolators. The LC network is described by

$$\Delta u(k) = \sum_{i=1}^{N} C_i\left(\phi^c(k)\rho_i(\psi(t))\right) = \sum_{i=1}^{N} \rho_i(\psi(t))g_i^T(w - f_i) \qquad (3.21)$$

$$u(k) = u(k-1) + \Delta u(k) \qquad (3.22)$$

It should be emphasized that, like the LM network, local models in LC networks do not have their own local states; instead, they share a 'common' state defined by the controller state $u(k)$ as defined in equation (3.21) and (3.22). Under the framework of the LC network and the LM network, some of the stability and robustness analysis for linear model predictive controllers (Robison and Clarke, 1991, Scokaert and Clarke, 1994, Soeterboek, 1992) could be extended to LM network based predictive control.

3.3 Case Study

In this section, two case studies are carried out on a highly nonlinear simulated process, the continuous stirred tank reaction (CSTR), as introduced in chapter 2. The CSTR is single-input, single-output process, where the input is the coolant flow rate $q_c(t)$ and the output is concentration of a product compound *C(t)*. The reaction that takes place to the compound is exothermic. It raises the temperature and reduces the reaction rate. The objective of the controller design is to control the product concentration *C(t)* by adjusting the induction of the coolant flow rate $q_c(t)$, which manipulates the temperature *T(t)*, and hence, control the product concentration.

A nonlinear PID controller based on the LM network model is designed first, which shows that LC networks provide a convenient approach to extend linear control theories to nonlinear system control compared with neural networks based PID controllers. Next, GPC (generalized predictive control) is applied to the CSTR process, based on the LC network framework. The online interpolation scheme in the LC network simplifies the design procedure and improves the performance globally. Then some issues related to the interpoaltion of local controllers are exposed in the comments.

3.3.1 PID Controllers

For many control applications, it is sufficient to use a standard PID controller. The purpose of this example is to illustrate the possible improvement obtained over the standard PID controller by designing local controllers based on the LM network model, which employs an integrator to eliminate the steady state errors.

Suppose a nonlinear PID controller of the form

$$u(t)=K_p(\psi(t))\left(e(t)+\frac{1}{T_i(\psi(t))}\int_{t0}^{t}e(t)dt+T_d(\psi(t))\frac{de(t)}{dt}\right) \tag{3.23}$$

is designed, where the gain K_p, integral time T_i and derivative time T_d are functions of the system's operating point specified by scheduling variables $\psi(t)$. In standard PID controllers, these functions are all constant, and there exists numerous design procedures, which guarantee that various stability, performance and robustness specifications are met. Of course, with the nonlinear PID controller above, the design problem using neural networks is much more difficult.

With respect to neural networks, generally, there are two different ways of applying a neural network to solve control-engineering problems. One approach is to use the neural networks to adjust the parameters of a conventional controller (Ruano et al., 1992, Yu and Lu, 1998). The other method is to use the neural network as a direct controller, which is trained to be the inverse of the nonlinear system (Chan et al., 1995, Chen and Chang, 1996, VanDoren, 2001). The first method involves emulating the thoughts of an expert control engineer by tweaking the tuning parameters according to empirical rules. Its application is limited and the computational load is normally intensive. The second method is an application of adaptive control, which has been used in industrial applications and brings some promising results. However, the selection of the network training data sets is not a trivial problem, and the computational load is highly intensive. So far, most of the approaches based on neural networks lose the simplicity of implementation, which is the most attractive feature of the original PID control approach.

However, the operating regime based approach offers an engineering friendly solution to this design problem. One can, for instance, design a number of standard linear PID controllers to meet the desired stability, performance and robustness criteria locally when the system is operating in neighborhoods of some selected operating points:

$$u^i(t) = K_p^i\left(e(t) + \frac{1}{T_i^i}\int_{t0}^{t} e(t)dt + T_d^i\frac{de(t)}{dt}\right) \tag{3.24}$$

Designing such local PID controllers is often simpler than approaching the nonlinear PID-controller design problem directly, even if a nonlinear dynamic model of the system exists. In addition, a weighting function for combining or switching between the local PID parameters is required. By using the LM network model, the validity function $\rho_i(\psi(t))$ can be employed. Then the final nonlinear PID controller is as follows:

$$u(t) = \sum_{i=1}^{N}\left(K_p^i\left(e(t) + \frac{1}{T_i^i}\int_0 e(t)dt + T_d^i\frac{d}{dt}e(t)\right)\right)\rho_i(\psi(t)) \tag{3.25}$$

with
$$K_p(\psi(t)) = \sum_{i=1}^{N} K_p^i\rho_i(\psi(t))$$

$$\frac{1}{T_i(\psi(t))} = \sum_{i=1}^{N}\frac{1}{T_i^i}\rho_i(\psi(t))$$

$$T_d(\psi(t)) = \sum_{i=1}^{N} T_d^i\rho_i(\psi(t))$$

Simulation results show two LC networks, designed based on the LM network models, for the CSTR plant. Controller C1 contains two local controllers. Controller C2 contains 5 local controllers. Both C1 and C2 are designed based on the LM network models illustrated in chapter two. The local PI/PID controller parameters were designed firstly using the Zeigler-Nichols ultimate cycle tuning rule (Zeigler and Nichols, 1942) for each local model; then the PID parameters are manually adjusted so that the $\pm 2\%$ settling time is about 0.5 min with a subsidiary specification being that the overshoot to a step command signal is less than 10%; then all the locally designed PID parameters were combined in the LC network structure to give a global nonlinear PID controller. One thing should be emphasied is that the desired performance target is only set to each local system. As for the global system, there is no specific control target could be set by using LC network. However, good global performance could be expected if expected target for local systems could be satisfied.

As well-known, there are many ways to set up the control target, settling time is one most commonly used in the time domain. It can be evaluated by examing the perforamnce figure. For the purpose to give a fair comparison between diference control methods, for instance, predictive control and pole placement, settling time is selected as the control target. In simulation, the following PID parameters are employed. The Local PID controller parameters for C1 are:

K_{p1} =1.7e3, T_{i1}=0.6396, T_{d1}=0.15

K_{p2}=2.1e3, T_{i2}=1.5832, T_{d2}=0

The Local PID controller parameters for C2 are:

K_{p1} =1.7e3, T_{i1}=0.6396, T_{d1}=0.15

K_{p2}=2.94e3, T_{i2}=01.0744, T_{d2}=0.1

K_{p3}=1.197e3, T_{i3}=2.27, T_{d3}=0

K_{p4}=6.945e3, T_{i4}= 3.472, T_{d4}=0

K_{p5}=4.458e4, T_{i5}= 0.535, T_{d5}=0

These controllers are, for most practical purposes, equivalent to the LC network used by Gawthrop (1995, 1996) and Hunt and Johansen (1997), where it is also shown that the LC network is superior to a single controller designed on thc basis of linearization about a single operating point. The global nonlinear PID controller is formulated by blending the local controllers through the interpolation system resulting from the LM network framework.

The overall performance of both the C1 and C2 nonlinear PID controlled systems are shown in figure 3-3, which covers set-point concentration *C(t)* changes from 0.03 mol/l up to 0.13 mol/l and includes both relatively over-damped (*C(t)*<0.1) and highly under-damped (*C(t)*>0.1) operating regimes. The overall closed-loop system, compensated by each of the controllers, shows good transient response when the set point *C(t)* changes from 0.03 mol/l up to 0.13 mol/I.

Generally, C2 (solid line) tracks the input changes faster and settles down quicker than C1 (dash-dotted line) does. When step changes are made from *C(t)*=0.1 to *C(t)*=0.13, i.e. so that the system remains in one local regime, C2 allows better closed-loop performance both in term of rise time and overshoot than C1 does. When step changes to *C(t)*=0.06, which is close to the representative operating points in C1($C_o^2 = 0.062 mol/l, T_o^2 = 448.7522K$, $q_{co}^2 = 90.0 l/\min$) and C2 ($C_o^3 = 0.05854 mol/l, T_o^3 = 450K, q_{co}^3 = 88.291 l/\min$), both of the LC networks show relatively good performance. Moreover, *C(t)*=0.13 is very close to the open loop unstable region of the CSTR process, in which there is only small closed-loop response overshoot using C2.

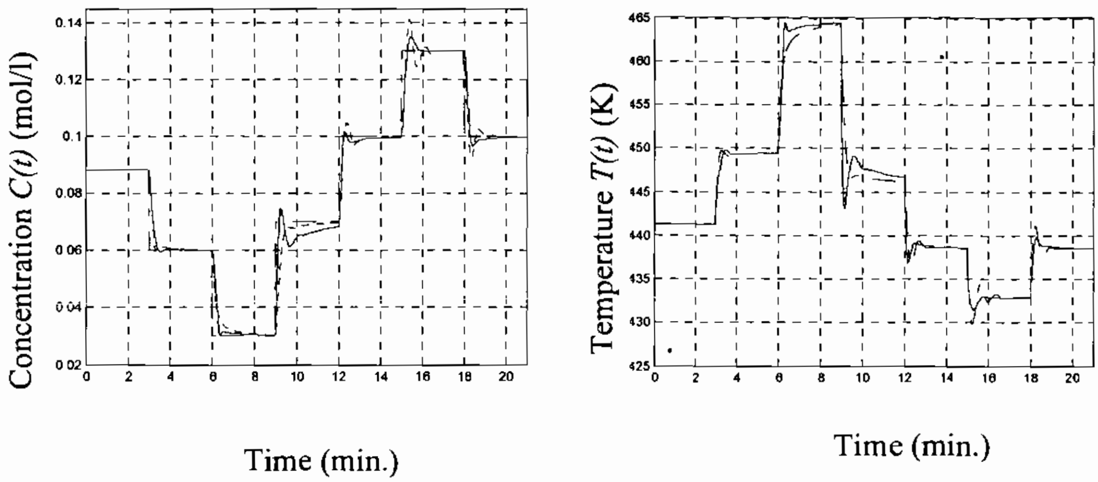


Figure 3-3. PID LC network controlled CSTR step responses

Dash-dotted line is the step changes; Dashed line is from C1 (with 2 local controllers), solid line is from C2 (with 5 local controllers)

Remark: The controller design procedure involves a trial-and-error process. Many simulations have been carried out, not only towards the desired local controller performance, but also towards a good global performance. A satisfactory local controller performance doesn't naturally guarantee a satisfactory global controller performance, even though all the locally designed PID controllers satisfy the design criteria locally.

It is important to emphasize that while LM and LC network pairs appear to offer a simple solution to the PID auto-tuning problem, it is not always easy to guarantee the global performance and analyse the properties of the overall controlled global system, especially during the operating regimes, where local controllers are blended to play the role. For instance, in Figure 3-3, the settling time for the concentration output at a level of 0.07 mol/l is greater than the local performance target of 0.5 minutes. These phenomena are closely related to the normalized weighting function, introduced in section 2.2.3. Furthermore, the interpolation technique may cause some side effects on the global stability and performance, which will be discussed in chapter four.

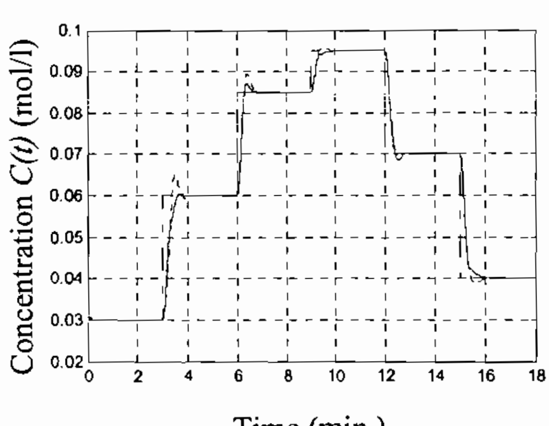
3.3.2 Predictive Controllers

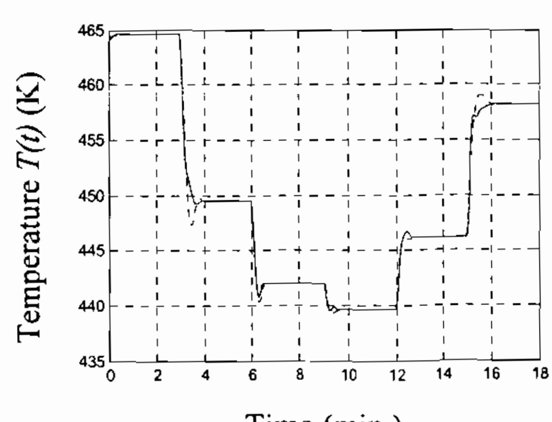
In this sub-section, examples are given to demonstrate the effectiveness of the LC network in a nonlinear control problem based on the generalized predictive control method, which optimizes the control strategy at every step based on a quadratic objective function over a finite time horizon. This approach doesn't involve the computationally intensive calculations normally required for nonlinear optimization. This is the advantage of incorporating the LM network into a model predictive control structure.

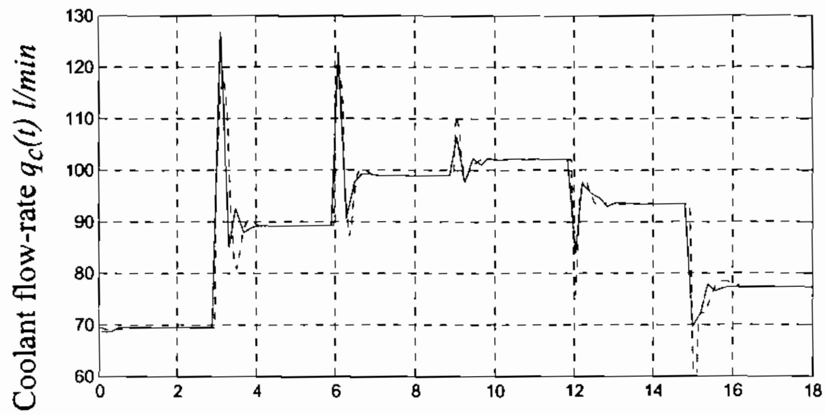
The simulation discusses two LC networks designed for the CSTR simulated process illustrated in chapter two. Controller C1 contains two local controllers, which are designed based on the LM network model with 2 local models in Chapter two. Controller C2 contains five local controllers, which are designed based on the LM network model with 5 local models in Chapter two.

In simulation, all the local controllers are designed with about 0.5 minutes settling time, and less than 10% overshoot. The prediction parameters H_P=5, H_m=1, H_c=1 and weights for control output in a quadratic objective function (equation 3.12) are λ = [5e-7, 1e-7] for C1 and λ=[5e-7, 5e-7, 5e-7, 3e-7, 1e-7, 1e-8] for C2 obtained by trial and error.

To check the performance of the LC network based on predictive control approach, a set of simulation experiments have been designed:







Time (min.)

Figure 3-4. Servo responses comparison for over-damped area

Dash-dotted line is the step changes; Dotted line is from C1 (with 2 local controllers); Solid line is from C2 (with 5 local controllers)

- Experiment E1 entails a set of step changes in a relatively over-damped area (*C(t)*<0.1). It intends to cover small and big, up and down step changes. Simulation results are shown in figure 3-4.

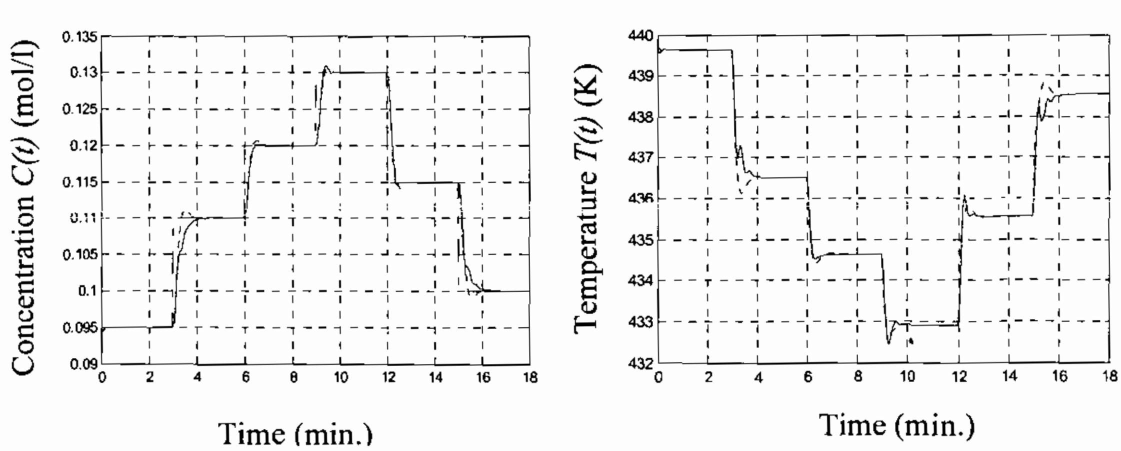
In experiment E1, the overall simulation result shows that C2 allows better performance than C1. When the system moves to the areas (say *C(t)*= 0.06 or 0.085) close to those special operating points $C_o^1 = 0.062 mol/l, T_o^1 = 448.7522K, q_{co}^1 = 90.0 l/\min$ (where a local controller is designed in C1), or $C_o^2 = 0.08506 mol/l, T_o^2 = 442K, q_{co}^2 = 98.899 l/\min$ (where a local controller is designed in C2), both the concentration output *C(t)* and the temperature output *T(t)* from C2 track the step change quicker and settle down faster than those from C1. C2 also performance better than C1 when the step changes to the set point, say *C(t)*= 0.095, which are some operating points away from those special points, where the local controllers are designed. In addition, $q_c(t)$ from C2 shows smaller and shorter oscillation than C1 does.

- Experiment E2 entails a set of step changes in a relatively under-damped area (*C(t)*>0.1), corresponding to highly dynamic area of CSTR. Simulation results are shown in Figure 3-5.

In experiment E2, the closed-loop simulation results using both C2 and C1 show reasonably good performance. From section 2.4.2, it is noticed that the LM network with two local models is oscillating when *C(t)*=0.11. However, in Figure 3-5, C1 shows reasonably good tracking ability. In contrast, C2 doesn't give as good performance as C1 does and presents slow tracking ability when *C(t)*=0.11, although the LM network with 5 local models shows better modelling accuracy at that working point.

When the effluent concentration *C(t)* moves upwards to 0.12mol/l and 0.13mol/l, C1 and C2 gives very close performance for the same unique local controller is employed from both C1 and C2 in this area. This local controller is designed based on the local model linearized at operating point $C(t) = 0.1298 mol/l$, $T(t) = 432.95K$, $qct = 110 l/\min$, when *C(t)*>0.115 (see figure 2-11 and figure 2-12). In addition, generally, the coolant flow rate $q_c(t)$ from C2 shows smaller and shorter oscillation than $q_c(t)$ from C1, although the difference is small.

From Figure 3-4 and Figure 3-5, it maybe noticed that, although C2 does show better control performance than C1, the performance difference between them is much smaller than in Figure 3-3, where the local controllers are experience-based PID controllers. In addition, it is obvious that the GPC based LC network facilitates much smaller overshoot and rise-time globally than the nonlinear PID controller implementation.



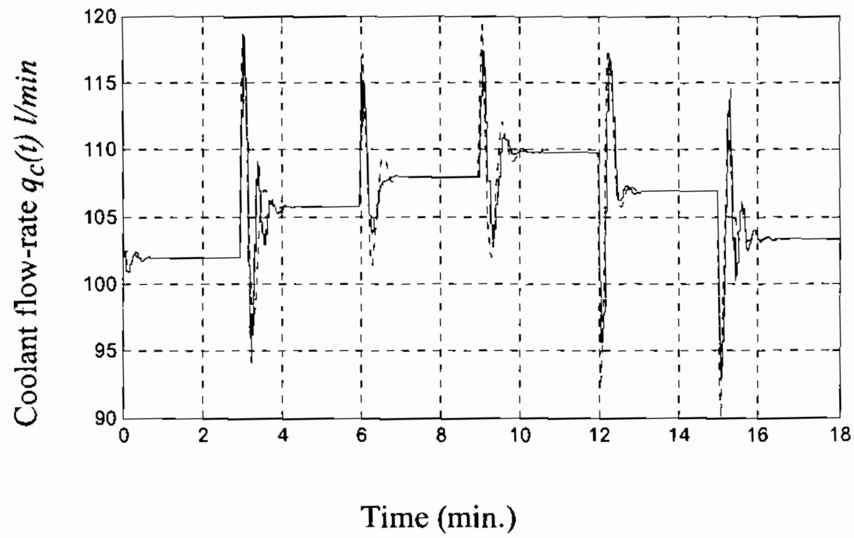
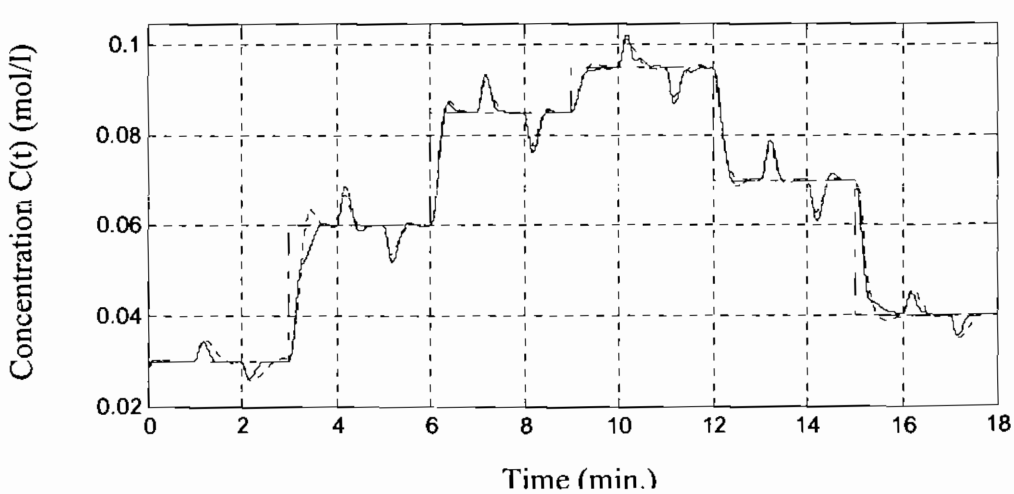


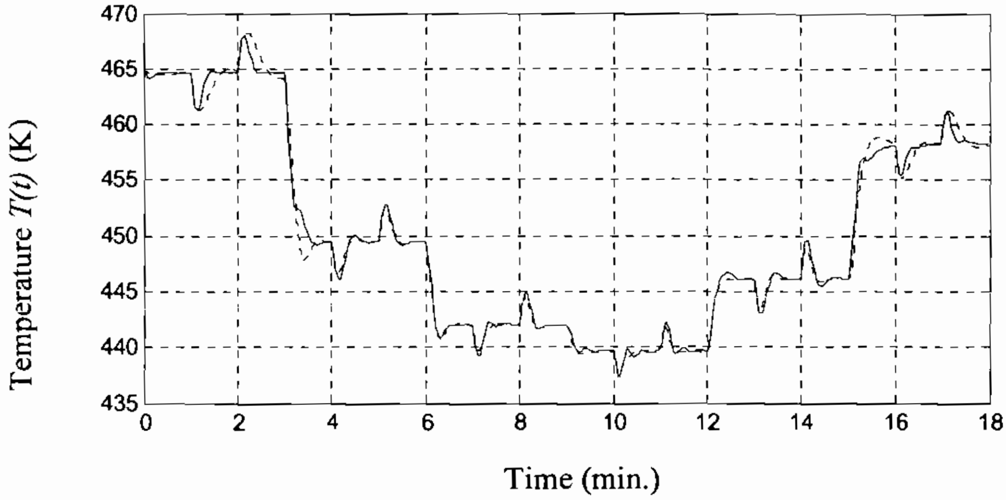
Figure 3-5. Servo response comparison for under-damped area

Dash-dotted line is the step changes; Dotted line is from C1 (with 2 local controllers); Solid line is from C2 (with 5 local controllers)

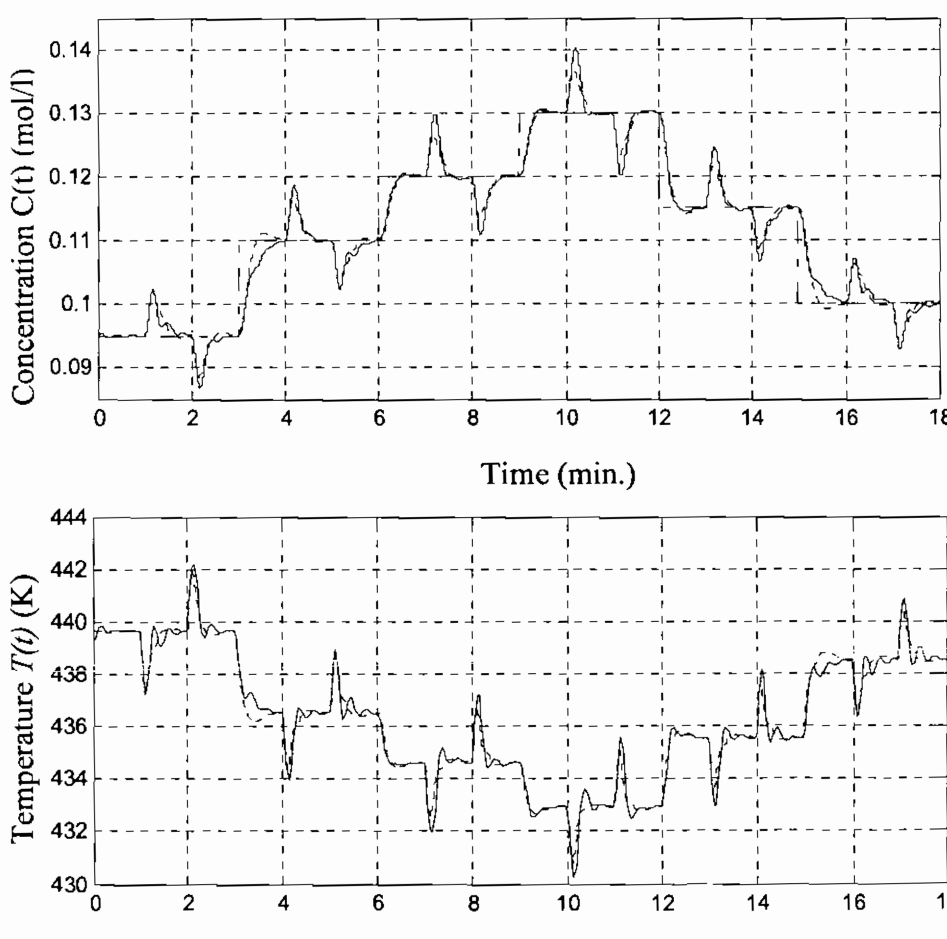
- Experiment E3 deals with the regulation performance of the closed-loop system controlled by C1 and C2. Simulation results are shown in figure 3-6.

Both the over-damped area and under-damped area have been tested. Introducing an impulse disturbance of 30*l/min* amplitude and 1*min* width to the system, the CSTR output goes back to the set point after short oscillations, when under the control of both C1 and C2 as shown in Figure 3-6.

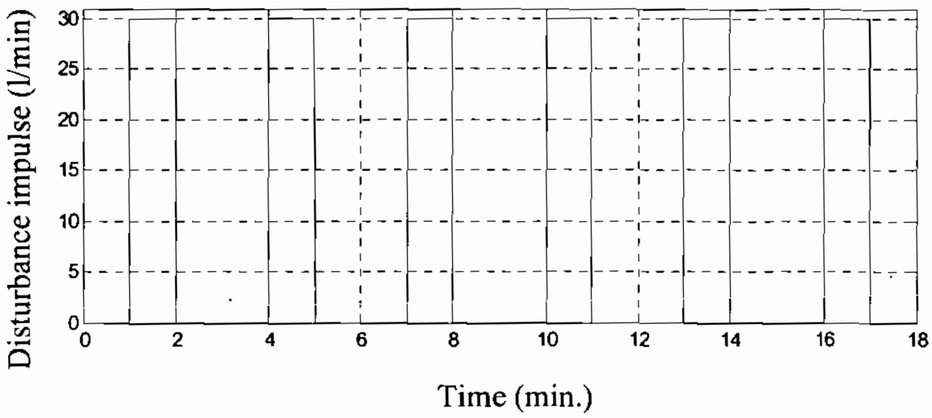




(a) Disturbance rejection comparison for under-damped area



(b) Disturbance rejection comparison for under-damped area



(c) Disturbance input

Figure 3-6. Comparison of regulation Performance of the controlled CSTR

Dash-dotted line is the step changes; Dotted line is from C1 (with 2 local controllers); Solid line is from C2 (with 5 local controllers)

Generally, the system outputs from both C1 and C2 are satisfactory. When *C(t)*=0.03 and 0.04, C2 goes back to the setpoint a bit quicker than C1 does. However, their performances are very close at most setpoints. C2, the LC network with 5 local controllers, doesn't present obvious advantages for disturbance rejection.

Remarks: The weighting factor $\lambda(j)$ in equation (3.12) for local GPC design is hard to choose such that the system behaves as desired, because $\lambda(j)$ is highly application dependent and must usually be determined by simulations in combination with the well-known and often-used trial-and-error method (Soeterboek, 1992). For local controller design, the trial-and-error method seems effective, but when the operating points moves, an adaptable $\lambda(j)$ should be expected. No knowledge is so far available to allow $\lambda(j)$ to be selected automatically. However, increasing the number of local controllers can reduce the requirement for selecting $\lambda(j)$ to some degree and lead to better global performance.

3.3.3 Comments

The results from the two application cases aforementioned clearly demonstrate the ability of the LC network to adapt its characteristics to the nonlinear plant. This is achieved through combining the local controllers between operating regimes by utilizing the interpolation technique resulted from the LM network model.

It is important to note that the local model is only used to design a local controller. The real meaning of the local model is that if the system operating point lies in the designed local regime, then the control of the system depends on the local model and the local controller. In this case, the global performance of the LC network highly depends on the performance of the local controllers. However, after introducing the 'divide and conquer' interpolation technique to blend the local controllers, the definition of 'local regime' is not highly transparent. In the operating regime between the representative operating points, co-operation of the local controllers is required rather than any local controller to play the role solely.

From the modeling point of view, the LM networks model with five local models gives a good approximation for the process based both on static and dynamic considerations; the LM network model with two local models, however, is able to approximate the dynamic properties of the CSTR simulated process, although the steady state errors are significant. From the control point of view, this static modelling inaccuracy is not a difficult issue, if the LM network can instantly capture the dynamics of the controlled system, assuming the system changes 'slowly' enough. Employing an integral action in local controller design can eliminate constant steady state errors to some degree. However, the integral action only asymptotically rejects the modeling error and disturbances in steady-state operation around the equilibrium point; the transient performance is affected when combining the local controllers together to control the process.

Properly increasing the number of local model and local controller pairs reasonably improves the global control performance, which benefits from better modeling accuracy, but increasing the number of local model and controller pairs also increases the difficulty of design due to the interpolation function. Generally, the controller design procedure involves a trial-and-error process. Simulations have been carried out, not only towards desired local controller performance, but also towards an expected global performance. A satisfactory local controller performance doesn't naturally guarantee a satisfactory global controller performance, even though all the locally designed controllers satisfy the design criteria locally. This phenomenon is a problem with the method. The blending of local controllers brings some new interesting issues, which are not transparent, but they influence the global performance robustness and the global stability, if the system change is not 'slow' enough (Leith and Leithead, 1999). This issue will be discussed in chapter four.

In summary, generally, introducing integrative actions in local controller design is able to overcome the steady state modelling errors to some degree in the application of LC/LM networks for nonlinear processes; properly increasing the local model and local controller pairs improves the controlled system performance, which benefits from the improved accuracy in steady state modelling. Moreover, the generalized predictive control application with the LC network brings on-line optimization techniques into the controller design process, which improves the global LC network controller performance.

One more thing worth mentioning is that, for the case of the control of the simulated CSTR process, both of C1 and C2 present good tracking and regulation performance, although C2 gives better simulation results. Simulations have also been implemented based on 3 and 4 local models and controllers, but no obvious performance improvement could be observed compared to C1. It is noted that the LM network built with two local models is good enough in this simulation case for the controller design if proper control techniques have been selected. From the engineering point of view, the smaller the number of the local models, the lower the overhead for the controller design.

3.4 Globally Designed Controller

3.4.1 Controller Formulation

The underlying assumption in the local modelling strategy of LM networks is that the plant to be modelled undergoes significant changes in operating conditions. The complexity of the overall model can be reduced by incorporating simpler models in each operating region. For example, local state-space and ARMAX linear models can be formed first and then interpolated to give global nonlinear state-space models (Priestly,1988) and NARMAX (nonlinear ARMAX) models (Johansen and Foss, 1993). This section considers the controller design based on the global LM network model.

By using the LM network, the state space description of the plant dynamics as in equation (2.26) can be rewritten as follows:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\rho)\boldsymbol{x}(t) + \boldsymbol{B}(\rho)\boldsymbol{u}(t) + \alpha(\rho)$$

$$y = \boldsymbol{C}(\rho)\boldsymbol{x}(t) \qquad (3.26)$$

where $\boldsymbol{A}(\rho) = \sum_{i=1}^{N} \rho_i(\psi(t))\boldsymbol{A}_i$, $\boldsymbol{B}(\rho) = \sum_{i=1}^{N} \rho_i(\psi(t))\boldsymbol{B}_i$,

$\alpha(\rho) = \sum_{i=1}^{N} \rho_i(\psi(t))\alpha_i$ and $\boldsymbol{C}(\rho) = \sum_{i=1}^{N} \rho_i(\psi(t))\boldsymbol{C}_i$.

ρ is the weighting vector; each value $\rho_i(\psi(t))$ is associated with a local model, and determines the validity of the operating regime for the current operating point defined by scheduling vector $\psi(t)$, which is associated with system states and system input and output. It should be noted that the above model is nonlinear in nature since the weighting functions are nonlinear and parameter-dependent in general. However, at any sampling instant, a linear (affine) model representation is formulated to approximate the system instant dynamic behaviour.

Recall the input-output representation of LM networks as in equation (2.29), and rewrite it as follows:

$$\hat{y}(k+1)=\phi_{ar}^{T}(k-1)\sum_{i=1}^{N}\rho_i(\psi(k))\theta_i+e(k)=\phi_{ar}^{T}(k-1)\Theta+e(k) \tag{3.27}$$

in which $\phi_{ar}^{T}(k-1)=\left[-y(k-1),\ldots,-y(k-n_y),u(k-n_k),\ldots u(k-n_k-n_u)\right]^{T}$,

$\theta_i=[a_{1i},\ldots,a_{n_y i},b_{0i},\ldots,b_{n_u i}]^{T}$,

$\Theta=[A_1,\ldots,A_r,B_0,B_1,\ldots,B_s]$.

The resulting model is still ARX in structure, with parameters A_i and B_i determined as follows:

$$B_i=\sum_{j=1}^{N}\rho_j(\psi(k))b_{ij} \text{ and } A_i=\sum_{j=1}^{N}\rho_j(\psi(k))a_{ij}.$$

Both A_i and B_i are parameter-dependent variables. Rewrite the equation (3.27) as follows:

$$\begin{aligned}\hat{y}(k+1)&=B_0(k)u(k)+B_1(k)u(k-1)+\cdots+B_s(k)u(k-s)+\cdots\\&\cdots-A_1(k)y(k)-A_2(k)y(k-1)-\cdots A_{r+1}(k)y(k-r)+e(k)\end{aligned} \tag{3.28}$$

The sequence *e(k)* contains unknown, unconstructed uncertainty, such as disturbances, unmodelled dynamics and modelling error.

At any sampling instant, a linear model is formulated to describe the instantaneous dynamics of the nonlinear system

$$A^{*}(q^{-1})\hat{y}(k+1)=B^{*}(q^{-1})u(k)+e(k) \tag{3.29}$$

in which $A^{*}(q^{-1}) = 1 + A_1^{*}q^{-1} + \cdots + A_{r+1}^{*}q^{-(r+1)}$,

$$B^{*}(q^{-1}) = B_0^{*} + B_1^{*}q^{-1} + \cdots + B_s^{*}q^{-s}.$$

At the current operating point, equation (3.29) is an instantly linearized model with ARX structure; a linear online controller can be easily designed from this model. Theoretically, any available linear controller design approach can be employed for the local controller design to formulate a globally effective LC network. In this section, the GPC method is employed for the local controller design. This kind of approach is called LMN-GPC in the book by Kouvaritakis and Cannon (2001).

There has been some research reported based on the idea of the global design approach, for example, internal model control (IMC) (Brown et al., 1997), dynamic matrix control (DMC) (Townsend et al., 1998) and generalized minimum variance (GMV) control (Brown and Irwin, 1999) control. With regard to linear model predictive control (MPC) applications for the control of nonlinear processes, one challenging problem is of guaranteeing the closed-loop stability, although there have been the emergence of some significant results: some of which use dual model predictions with terminal penalties and control Lyapunov functions, positively invariant sets and terminal stability constraints; others deploy feedback linearisation or differential flatness (Kouvaritakis and Cannon, 2001).

Employing a LM network model for controller design has potential to build a controller with guaranteed performance and global stability. There now exists a whole range of techniques for the application of multiple models: some use a common Lyapunov function (Wang et al, 1996); some make use of a set of Lyapunov functions with some restriction to the weighting functions (Daafouz et al., 2002); some make use of the stability analysis of polytopic description, in which it is needed to find out that all the system roots are in the unit circle (Farinwata et al., 2000, Filev 1996). The global stability would be one of the important issues for future work.

In addition, the LM network formulates a one parameter-dependent linear structured model to approximate the nonlinear process by blending a set of linear local models at any instant. The LM network transforms a nonlinear system into a user-specified and more transparently described LPV system. One advantage of using the LM network for controller design is that it avoids techniques which require nonlinear optimisation when using a nonlinear neural network model of the process. Moreover, there exists important contributions for the closed-loop stability of LPV systems (Lu and Arkun, 2000, Cuzzola et al., 2002, Rong, 2002, Wang and Balakrishnan, 2002), which utilise the combination of MPC and linear matrix inequality (LMI) .

3.4.2 Case Study

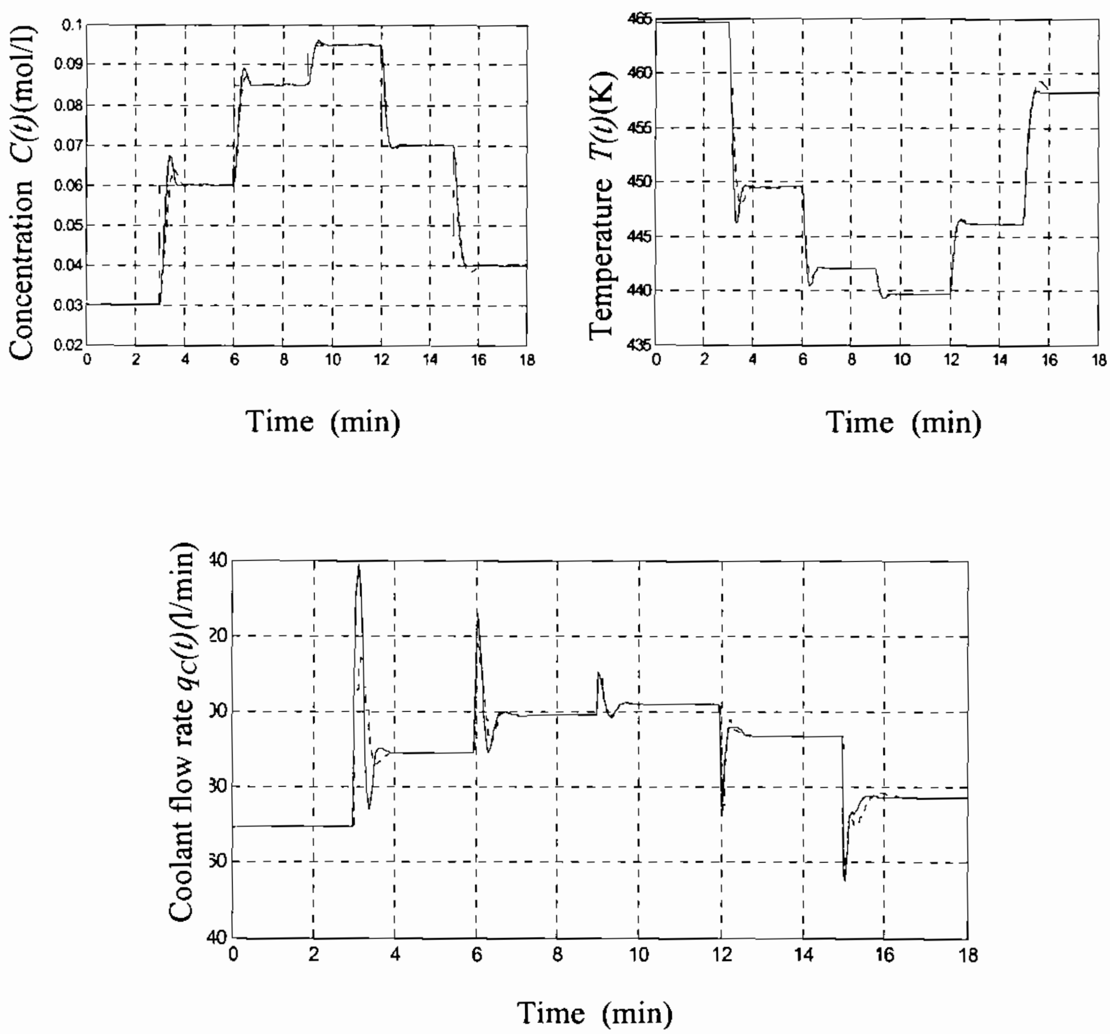
The GPC algorithm as introduced in section 3.2.3 is utilized to examine the performance of the global design approach. To ensure stability, it is assumed that the finite horizon length is chosen long enough such that the objective function cost decreases as time increases.

Consider the CSTR simulated process presented earlier. The controller design objective is to instantaneously design a linear predictive controller that drives the concentration of the effluent to track a known trajectory.

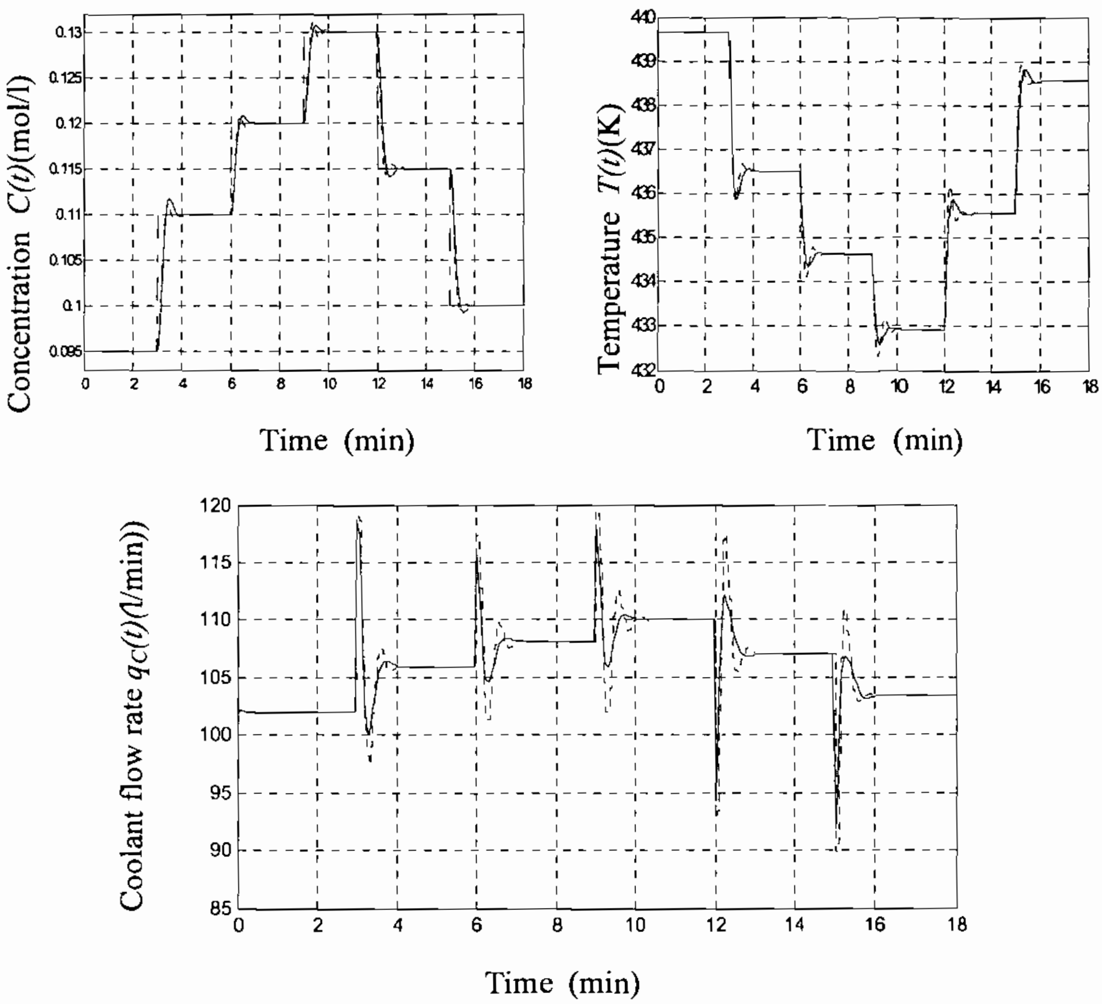
The simulation concerns two controller design for the CSTR simulated process based on the LM network models. Controller C1 is designed based on the LM network model with two local models; Controller C2 is designed based on the LM network model with five local models. Each local controller in both C1 and C2 follows the same design parameters as in section 3.3.2, i.e., H_p=5, H_m=1, H_c=1, but the weights λ for control output in the quadratic objective function is uniquely defined as λ = 5e-7. A set of simulation experiments have been carried out:

- Experiment E1 entails a set of step changes in a relatively over-damped area ($C(t)$<0.1) and underdamped area ($C(t)$<0.1). The simulation results are shown in Figure 3-7.

In Figure 3-7, the results show that there is little advantage in using 5 local models compared to using 2 lcoal modes based on the globally designed LC network approach. For example, the overshoot and settling time values show only minor differences when comparing the effect of C1 and C2 at most of the levels in both under-damped and over-damped areas.



(a) Servo responses comparison between C1 and C2 for over damped area

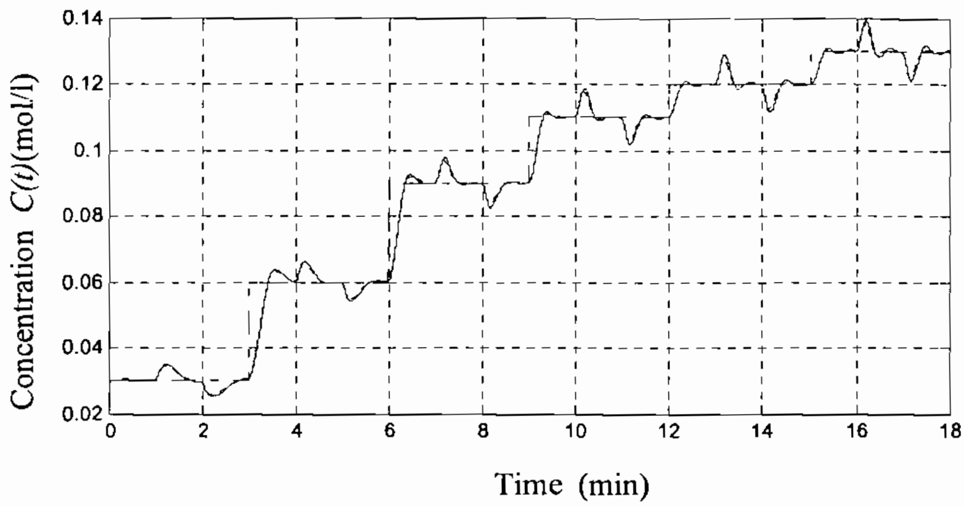


(b) Servo response comparison between C1 and C2 for under-damped area

Figure 3-7. Servo response comparison from the globaly designed LC network between C1 and C2
Dash-dotted line is the step changes; Dotted line is from C1 (with 2 local controllers); Solid line is from C2 (with 5 local controllers).

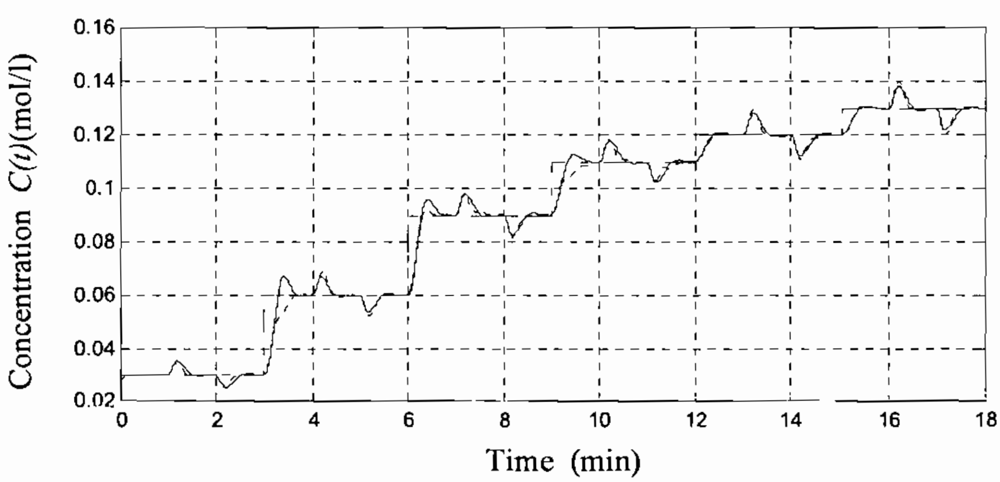
Remarks: The selection of $\lambda(j)$ plays an important role in the controller performance and stability for the global controller design approach, as in the gain scheduled LC network. It is ideal to have a desired $\lambda(j)$ at any instant of sampling time. However, there is no method available so far to automatically allow the selection of $\lambda(j)$. In simulation, the same values of $\lambda(j)$ were employed as the ones in the gain scheduled LC network controller, which other simulations have shown to allow better performance than a fixed $\lambda(j)$, over the operating range.

- Experiment E2 compares the disturbance performance from the gain-scheduled LC network and the globally designed controller as shown in Figure 3-9 for the concentration output *C(t)*. The same disturbance is employed here as in Figure 3-6 (c).



(a) Comparison of C1 concentration output with disturbance

Solid line is from globally designed controller and dashed line is from gain-scheduled LC network based on the LM network with 2 local models.



(b) Comparison of C2 concentration output with disturbance

Solid line is from globally designed controller and dashed line is from gain-scheduled LC network based on the LM network with 5 local models.

Figure 3-8. Comparison of disturbance performance from the gain-scheduled LC network and globally designed controller

In figure 3-8, the CSTR concentration outputs go back to the set point after a short oscillation under the control of any designed nonlinear MPC. The disturbance performances from the globally designed controller shows very little difference to those from the gain-scheduled LC networks for both C1 (Figure 3-8 (a)) and C2 (Figure 3-8 (b)). From the point of view of disturbance rejection, the globally designed controller works as well as the gain-scheduled LC network does.

Generally, in Figure 3-7, the closed-loop servo response results using either controller C1 or C2 are very close and indicate reasonable correspondence with the local design specifications. When adding the disturbances as shown in figure 3-8, the system output also shows good robustness. In summary, simulation results for the CSTR process illustrate the potential offered by controller design based on the global LM network model in terms of tracking and disturbance rejection.

3.5 Concluding Remarks

This chapter classified the reported approaches for controller design based on local model networks into two categories. The gain scheduled LC network is composed of a fixed number of local controllers that are designed offline based on corresponding locally valid linear models. The global controller design method is to design a controller online, based on an instant linear representation model resulting from LM networks.

For the gain-scheduled LC network approach, the local models are only used to design local controllers. The real meaning of the local model is that if the system operating point lies in the designed local regime, the local controller satisfies the local control target. However, after introducing the interpolation technique to combine the local controllers, the global performance normally is not able to meet the local control target.

In summery, both of the controller design approaches illustrate the simplicity and practicality of the LM network application in the control of nonlinear systems. In the case study, the combination of the LM network and GPC enables the global controller to perform satisfactory across the entire operating range. Moreover, one advantage of these approaches is that they avoid time-consuming numerical optimization methods, which often happen in conventional nonlinear model based predictive control.

One more thing is that for the nonlinear process CSTR, the LM network built with two local models is good enough in this simulation case for the controller design provided the proper control techniques have been selected. From the engineering point of view, the smaller the number of the local models, the lower the overhead for the controller design.

Under the assumption that the system changes 'slowly' enough, an integrator could be introduced to eliminate the modelling errors to zero. Introducing an integrative action and online predictive control drives the closed-loop system output to the set point with good speed and accuracy. However, if the system does not change 'slowly' enough, the offset term in the affine model, which is actually nonlinear and time varying after interpolation, shouldn't be simply taken as constant. The blending technique brings some new interesting issues, which influence the global stability and the global performance robustness at some stage. Later chapters attempt to present a systematic analysis and design methodology for the control of a class of nonlinear systems based on interpolation of linear subsystems.

Chapter 4

Synthesis & Analysis of LM/LC Networks

4.1 Introduction

Local model/controller (LM/LC) networks may be considered as the blending of relatively simple local subsystems to model and control overall complex nonlinear systems. Theoretically, from the modelling point of view, the finite set of linearizations about a finite number of points (equilibria and transient points) can be used to accurately approximate dynamic linearizations about arbitrary trajectories using a blended multiple model structure (Johansen et al., 1998a). Linear (homogenous) local model (LLM) networks inherit many valuable properties, but these networks with limited numbers of local models could often result in poor global representations of the nonlinear plant, due to the side effects resulting from the interpolation techniques (McLoone, 2000, McLoone et al. 2001). In contrast, affine (inhomogeneous) local model (ALM) networks improve the modelling accuracy of LLM networks significantly with the benefit of the extra offset term introduced in the ALM networks.

From the control point of view, LC networks adopt the same idea as Takagi-Sugeno (TS) fuzzy logic control systems, whose attraction is that a nonlinear design task is simplified to linear design problems by first decomposing the problem into a number of linear sub-problems solvable by established methods and recombining, in some appropriate manner, the resultant collection of linear designs to obtain the required nonlinear design.

The inherent nonlinearity in the local model (LM) network structure makes it possible to utilize linear models for the modelling of nonlinear processes with certain accuracy; the inherent nonlinearity in the local control (LC) network structure provides promising performance by simply using well-developed linear control approaches. However, the inherent nonlinearity also brings some disadvantage of the approach, for recent research found that the blending (interpolation/scheduling) procedure could play important roles in the stability, performance and robustness of the interpolation system (Murray-smith et al., 1999, Leith and Leithead, 1998, 2000).

It has become evident that many basic issues remain to be further addressed. Many requirements are common to the controller design based on local model representations of nonlinear dynamic systems; such controller design methods include gain scheduling, TS fuzzy logic systems, and statistical mixture control systems. These methods have attracted a rapidly growing interest in recent years. Stability analysis and systematic design are certainly among the most important issues for the controller design problem.

Moreover, the linear local model (LLM) has its equilibrium centred at the origin x=0. In contrast, the ALM is inhomogeneous and has a constant offset term. The literature review for these two approaches shows unevenly distributed interest, although the ALM blending system has been widely applied in the modelling of nonlinear systems. Most of the research work reported has been devoted to analysis of linear systems (Wang et al., 1996, Cao et al., 1998, Daafouz et al., 2002), although there are several interesting reports on the use of affine fuzzy systems or piecewise affine systems (Johansen et al., 1998a, Johansson et al., 1999, Kim and Kim, 2001,2002). The novelty of this chapter is the stability analysis of ALM blending systems by using ultimate boundedness analysis.

In chapter 3, we presented gain-scheduled LC networks, in which the offset terms resulting from the affine local models were considered as 'constants' under the assumption that the system changes slowly enough. This chapter presents a further step towards a systematic design methodology based on LM networks, which guarantees the stability of the closed-loop system. It employs quadratic Lyapunov functions to analyze the influence of offset terms on stability issues in affine blending systems. It deals with the offset terms, which partially represent the dynamics of nonlinear systems, in detail rather than simply dealing with the offset terms as 'constants' in the proposed gain-scheduled LC networks.

The chapter is organized as follows. In section 4.2, the stability conditions for inhomogenous ALM blending systems are discussed based on the analysis of the stability of LLM homogenous systems. Section 4.3 proposes a state feedback controller in the LC network framework with the aim of providing an overall stable closed-loop system with guaranteed stability and improved performance. Section 4.4 concludes the chapter with comments on the approach.

4.2 Stability Analysis of Homogenous Blending Systems

Consider the SISO nonlinear system as in equation (2.3). By a blended local model structure we understand a dynamic model of the form given by equation (2.1), and rewrite it as follows:

$$\dot{x} = \sum_{i=1}^{N} \rho_i(x,u) f_i(x,u) \tag{4.1}$$

where state vector is x, input is u, the model $f_i(:,:)$ is one of N vector functions of the states and the input, and is valid in a region defined by the scalar validity function ρ_i, which is, in turn, a function of the above variables. Typically, the local models f_i are chosen to be of the affine form $f_i(x,u) = A_i x + B_i u + d_i$, resulting in constituent dynamic systems $\sum_i$ given by,

$$\dot{x} = A(x,u)x + B(x,u)u + d(x,u) \tag{4.2}$$

where $A(x,u) = \sum_{i=1}^{N} \rho_i(x,u) A_i$, $B(x,u) = \sum_{i=1}^{N} \rho_i(x,u) B_i$ and $d(x,u) = \sum_{i=1}^{N} \rho_i(x,u) d_i$.

Several limitations of the multiple model approach are reviewed in (Shorten et al. 1999). These limitations can be summarized as the difficulties, or confusion, in understanding the meaning of the multiple models, which, to a high degree, resulted from the blending procedure. Such an understanding may be vital when the model is being used for designing a control system. This section concentrates on stability issues resulting from the blending framework. In particular, the stability behaviour associated with ALM networks will be investigated.

4.2.1 Stability Issues Resulting from Blending

The LC and LM network pairs concern the incorporation of N locally valid subsystems. Assuming that all the local closed-loop subsystems are stable, a question naturally arises as to whether the overall global closed-loop system is stable. The answer is no, in general. Although the stability, performance and robustness properties of each linear local model controller are well understood and can be analyzed using standard tools, such as the Bode plot and Nyquist plot for each fixed operating point, these local properties do not naturally and necessarily lead to guaranteed global properties (Wang, 1996, Rong, 2002, McConley et al., 2002).

One serious problem that the blending procedure could cause is the instability of the overall system, although each subsystem is locally stable (Wang et al. 1996). Thus stability issues should be taken into consideration when selecting validity functions and local models, and in the controller design of the blending system. How to systematically select validity functions, local models and approaches for controller design to meet the required overall system stability is not clear so far. Most of the time, a trial-and-error procedure has been used (Tanaka and Sugeno, 1992, Wang et al. 1996).

Hunt and Johansen (1997) proposed one sufficient but not necessary condition to guarantee the overall stability for gain scheduling systems, which is based on the analysis of the effect of modeling errors. However, the condition is mainly of a qualitative nature, as no bounds on performance, robustness or design parameters are provided. The objective of the work in this section of the chapter is to determine the bounds for the stability conditions of the blending affine systems by using quadratic Lyapunov functions.

4.2.2 Quadratic Stability of Blending System

4.2.2.1 Linear Local Models

For a blending system with linear local models, whose offset terms fade to zero, a common sufficient condition for the stability is given by the Lyapunov function. Recall the state space representation of the LM network as given by equation (4.2); the homogeneous system corresponding to (4.2) is

$$\dot{x} = \sum_{i=1}^{N} \rho_i A_i x \tag{4.3}$$

where the validity function $1 \geq \rho_i \geq 0$ and $\sum_{i=1}^{N} \rho_i = 1$. Each linear component $A_i x(t)$ is called a subsystem. The sufficient conditions for ensuring stability of equation (4.3) are usually formulated in Theorem 1 (Tanaka and Sugeno, 1992):

Theorem 1: The equilibrium point of a system (4.3) is globally exponentially stable if there exists a common positive definite matrix P such that

$$A_i^T P + P A_i < 0 \,, \; i = 1,2,\cdots,N \tag{4.4}$$

i.e., a common P has to exist for all subsystem to guarantee the overall stability. In this case, the nominal global system has a uniformly exponentially stable equilibrium point at the origin. This theorem reduces to the Lyapunov stability theorem for linear systems when N=1.

The stability condition of Theorem 1 is derived using a quadratic function $V(x) = x^T Px$. If there exists a $P>0$ such that the quadratic function proves the stability of system (4.3), system (4.3) is also said to be quadratically stable and the function V is called a quadratic Lyapunov function. Theorem 1 thus presents a sufficient condition for quadratic stability of system (4.3).

Checking the stability of system (4.3) has long been recognized to be difficult for there is a lack of a systematic procedure to find a common positive definite matrix P. Solving this problem requires the answer to two questions: Is there a common quadratic Lyapunov function that exists? How can a common quadratic Lyapunov function be determined?

Deriving sufficient conditions under which exponential stability will be assured has been investigated by a number of authors. Narendra and Balakrishnan (1994b) introduce 'commutativity' to assure the existence of a common Lyapunov function; however, the converse of the approach does not hold in general i.e. if there is no such commuting Lyapunov function found for the overall system, it doesn't mean that the overall global system is not stable; thus, the utilization of other approaches to look for the common P matrix, if it exists, is needed. Shorten and Narendra (1999) presented the necessary and sufficient conditions for the existence of a common quadratic Lyapunov function for two stable second-order linear systems. Subsequently, Shorten and Narendra (2002) extended the approach to check the existence of a common quadratic Lyapunov function for a finite number of stable second-order linear systems. Recently, the authors generalized the above results in (Shorten et al., 2003).

To determine the common quadratic Lyapunov function, most of the time a trial-and-error procedure has been used (Tanaka and Sugeno, 1992). In the literature, since the middle of the 1990s, there has been a rapidly growing interest in finding out the common Lyapunov function P by solving a convex optimization problem using linear matrix inequality (LMI) approaches (Boyd et al., 1994). A very important property of these approaches is that the stability condition of theorem 1 is expressed in LMI form (Wang et al., 1996, Daafouz et al., 2002). To check stability, which means to find a common positive quadratic Lyapunov function P, or to show that there is no such common P that exists for the system, converts to a problem of solving LMI functions.

Numerically, LMI problems can be solved efficiently by means of some powerful tools available in the mathematical programming literature, like the Matlab LMI toolbox.

4.2.2.2 Affine Local Models

For a blending system with linear local models (each of which has a stable node with an equilibrium point centered at the origin), the global system has its equilibrium point centered at the origin. In contrast, the affine local model allows its equilibrium point to be close to, but not centered at the origin (Murray-Smith et al., 1999), because the offset term in each affine local model doesn't fade to zero, but is instead a constant. Thus, the origin $x=0$ may not be the equilibrium point of the blended system. We can no longer study the stability of the system with the origin as an equilibrium point, nor should we expect the solution of the offset term to approach the origin as $t \to \infty$.

One possibility is to consider the offset term $d(x,u,t)$ in equation (4.2) as a non-vanishing perturbation. It is hoped that if the 'perturbation term' $d(x,u,t)$ is small in some sense, then $x(t)$ will be ultimately bounded by a small bound; that is, $\|x(t)\|$ will be small for sufficiently large t.

Consider the homogeneous system as in equation (4.2) and rewrite it as follows:

$$\dot{x} = \sum_{i=1}^{N} \rho_i A_i x + \sum_{i=1}^{N} \rho_i d_i \tag{4.5}$$

Note that equation (4.3) is termed a 'nominal system' and equation (4.5) is termed a 'perturbed system' for convenience.

Suppose the nominal system (4.3) has a uniformly asymptotically stable equilibrium point at the origin, what can we say about the stability behaviour of the perturbed system (4.5)? A natural approach to address this question is to use the Lyapunov function for the nominal system as a Lyapunov function candidate for the perturbed system. The new element here is that the 'perturbation term' will not vanish at the origin, i.e. the origin will not be an equilibrium point of the perturbed system. Therefore, the problem can no longer be studied as a question of the stability of equilibria. The best that can be hoped for is that the perturbation term, bounded by a small bound $\|x(t)\|$, will be small for sufficiently large t.

From Theorem 1, it is known that if there is a common positive definite P existing for all the local linear models, then $V(t,x)=x^T Px$ is a Lyapunov function of the global blending system (4.3). The conditions to ensure the global stability of system (4.5) are more complicated, as more analysis needs to be performed. As a special application case of the ultimate boundedness theory (Khalil, 1992), lemma 1 is given below:

Lemma 1: Let x=0 be an equilibrium point for the blended system in equation (4.3), where A_i is Hurwitz. Let $V(x)$ be a Lyapunov function of the nominal system. Then V satisfies the inequalities:

$$c_1\|x\|_2^2 \le V(x) \le c_2\|x\|_2^2 \tag{4.6}$$

$$\frac{\partial V}{\partial x}\sum_{i=1}^{N}\rho_i A_i x \le -c_3\|x\|_2^2 \tag{4.7}$$

$$\left\|\frac{\partial V}{\partial x}\right\|_2 \le c_4\|x\|_2 \tag{4.8}$$

for some positive constants c_1, c_2, c_3 and c_4, where

$$c_1 = \lambda_{\min}(P),\ c_2 = \lambda_{\max}(P),\ c_3 = \min_{i=1\cdots N}(\lambda_{\min}(Q_i)) \text{ and } c_4 = 2\lambda_{\max}(P)$$

Proof: Assuming that the Lyapunov function is defined as $V(x)=x^T Px$, P being the common positive definite matrix, then

$$\lambda_{\min}(P)I \le P \le \lambda_{\max}(P)I \quad \Rightarrow \quad x^T\lambda_{\min}(P)x \le V(x) \le x^T\lambda_{\max}(P)x$$

$$\Rightarrow \quad \lambda_{\min}(P)\|x\|_2^2 \le V(x) \le \lambda_{\max}(P)\|x\|_2^2$$

This result proves (4.6).

$$\frac{\partial V}{\partial x}\sum_{i=1}^{N}\rho_i A_i x = x^T\left(P\sum_{i=1}^{N}\rho_i A_i + \sum_{i=1}^{N}\rho_i A_i^T P\right)x = -x^T\sum_{i=1}^{N}\rho_i Q_i x$$

$$\leq -\sum_{i=1}^{N}\rho_i \lambda_{\min}(Q_i)\|x\|_2^2 \leq -\min_{i=1\cdots N}(\lambda_{\min}(Q_i))$$

This result proves (4.7).

$$\left\|\frac{\partial V}{\partial x}\right\|_2 \leq \left\|2x^T P\right\|_2 \leq 2\|P\|_2\|x\|_2 \leq 2\lambda_{\max}(P)\|x\|_2$$

This completes the proof of (4.8). Now we introduce some special scalar functions and Theorem 2 (Khalil, 1992) that will help to characterize and study the stability behavior of the blending affine local model systems.

Definition 4.1: A continuous function $\alpha:[0,a)\rightarrow[0,\infty)$ is said to belong to the class K function if it is strictly increasing and $\alpha(0)=0$. It is said to belong to the class K_∞ function if $a=\infty$ and $\alpha(r)\rightarrow\infty$ as $r\rightarrow\infty$.

Definition 4.2: A continuous function $\beta:[0,a)\rightarrow[0,\infty)$ is said to be a class KL function if for each fixed s, the mapping $\beta(r,s)$ belongs to class K with respect to r, and for each fixed r the mapping $\beta(r,s)$ is decreasing with respect to s, and $\beta(r,s)\rightarrow\infty$ as $r\rightarrow\infty$.

Theorem 2: Let $D=\{x\in R^n \mid \|x\|<r\}$ and $f:[0,\infty)\times D\rightarrow R^n$ be piecewise continuous in t and locally Lipschitz in x. Let $V:[0,\infty)\times D\rightarrow R$ be a continuous differentiable function such that

$$\alpha_1(\|x\|)\leq V(t,x)\leq\alpha_2(\|x\|) \tag{4.9}$$

$$\frac{\partial V}{\partial t}+\frac{\partial V}{\partial x}f(t,x)\leq-\alpha_3(\|x\|), \forall\|x\|\geq\mu>0 \tag{4.10}$$

$\forall t \geq 0$, $\forall x \in D$, where $\alpha_1(\cdot)$, $\alpha_2(\cdot)$, and $\alpha_3(\cdot)$ are class K functions defined on $[0, r)$ and $\mu < \alpha_2^{-1}(\alpha_1(r))$. Then, there exists a class *KL* function $\beta(\cdot,\cdot)$ and a finite time t_1 (dependent on $x(t_0)$ and μ) such that

$$\|x(t)\| \leq \beta(\|x(t_0)\|, t - t_0), \ \forall t_0 \leq t \leq t_1 \qquad (4.11)$$

and $$\|x(t)\| \leq \alpha_1^{-1}(\alpha_2(\mu)), \ \forall t \geq t_1 \qquad (4.12)$$

$\forall \|x(t_0)\| < \alpha_2^{-1}(\alpha_1(r))$. Moreover, if all the assumptions hold with $r = \infty$, that is, $D = R^n$, and $\alpha_1(\cdot)$ belongs to class K_∞, then inequalities (4.11)-(4.12) hold for any initial state $x(t_0)$. Furthermore, if $\alpha_i(r) = k_i r^c$, for some positive constants k_i and c, then $\beta(r,s) = kr\exp(-rs)$ with $k = (k_2/k_1)^{1/c}$ and $r = (k_3/k_2 c)$.

Inequalities (4.11)-(4.12) show that $x(t)$ is uniformly bounded for all $t \geq t_0$. They also show that $x(t)$ is uniformly ultimately bounded with an ultimate bound $\alpha_1^{-1}(\alpha_2(\mu))$. It is significant that the ultimate bound is a class K function of μ, because the smaller the value of μ, the smaller the ultimate bound. As $\mu \to 0$, the ultimate bound approaches zero.

Based on theorem 2, a lemma (Lemma 2) is developed for the analysis of the blending of affine local models, when the origin of the nominal system is exponentially stable.

Lemma 2: Let x=0 be an exponentially stable equilibrium point of the nominal system. Let *V(x)* be a Lyapunov function of the nominal system and for some positive $0 < \theta < 1$, then the solution of the perturbed system *x(t)* satisfies:

$$\|x(t)\| \leq k\exp[-r(t - t_0)]\|x(t_0)\|, \ \forall t_0 \leq t < t_1$$

and $$\|x(t)\| \leq b, \forall t \geq t_1$$

for some finite time t_1, where $k = \sqrt{c_2/c_1}$, $r = \dfrac{(1-\theta)c_3}{2c_2}$, $b = \dfrac{c_4}{c_3}\sqrt{\dfrac{c_2}{c_1}}\dfrac{\delta}{\theta}$

Proof: Assume $V(x)$ is a Lyapunov function candidate for the perturbation system (4.7). The derivative of $V(x)$ along the trajectories of (4.7) satisfies

$$\dot{V}(x) = x^T P\dot{x} + \dot{x}^T Px$$

$$= x^T P\left(\sum_{i=1}^{N}\rho_i A_i x + \sum_{i=1}^{N}\rho_i d_i\right) + \left(\sum_{i=1}^{N}\rho_i A_i x + \sum_{i=1}^{N}\rho_i d_i\right)^T Px$$

$$= x^T\left(P\sum_{i=1}^{N}\rho_i A_i + \sum_{i=1}^{N}\rho_i A_i^T P\right)x + 2x^T P\sum_{i=1}^{N}\rho_i d_i$$

$$\leq -c_3\|x\|_2^2 + c_4\|x\|_2 \max_{i=1\cdots N}(|d_i|)$$

(From Lemma 1 and $0 \leq \rho_i \leq 1$, $\sum_{i=1}^{N}\rho_i = 1$)

$$= -(1-\theta)c_3\|x\|_2^2 - (\theta c_3\|x\|_2 - c_4\delta)\|x\|_2, \quad \delta = \max_{i=1\cdots N}(|d_i|)$$

$$\leq -(1-\theta)c_3\|x\|_2^2 \quad \forall \|x\|_2 \geq c_4\delta/c_3\theta$$

The application of Theorem 2 completes the proof.

Lemma 2 shows the effect, from the offset term, of affine models on the property of blended systems. This result demonstrates that if nominal system (4.3) is exponentially stable with respect to the origin, then the corresponding perturbed system (4.5) is uniformly bounded with ultimate bound b. Moreover, note that the ultimate bound b is proportional to the upper bound on the perturbation $\delta = \max_{i=1\cdots N}(|d_i|)$. The ultimate bound can be viewed as a robustness property of nominal systems having exponentially stable equilibria at the origin, because it shows that arbitrarily small (uniformly bounded) perturbations will not result in large steady-state deviations from the origin.

4.3 Stabilization via Feedback Control

The standard state feedback stabilization problem for the system in equation (2.3) is the problem of designing a feedback control law, $u = \gamma(x)$, such that the origin x=0 is an asymptotically stable equilibrium point of the closed-loop system $\dot{x} = f(x, \gamma(x))$. In a typical control problem, there are additional goals for the design, like meeting requirements on the transient response or certain constraints on the control input, but the discussion in this section is mainly concentrated on the basic problem of stabilizing the blending affine systems.

Naturally, the feedback stabilization problem is much simpler when the system is linear:

$$\dot{x} = Ax + Bu$$

In this case, the state feedback control $u = kx$ preserves the linearity of the open-loop system, and the origin of the closed-loop system described by $\dot{x} = (A + Bk)x$ is asymptotically stable if and only if the matrix $A + Bk$ is Hurwitz. Thus, the feedback stabilization problem reduces to a problem of designing a matrix k to assign the eigenvalues of $A + Bk$ in the open left-half complex plane. Linear control theory (DeCarlo, 1989) confirms that the eigenvalues of $A + Bk$ can be arbitrarily assigned provided that the pair (A, B) is controllable. Even if some eigenvalues of A are not controllable, stabilization is still possible provided the uncontrollable (open-loop) eigenvalues of A will be (closed-loop) eigenvalues of $A + Bk$.

For a general nonlinear system, the problem is more difficult and less well understood. The most practical way to approach the stabilization problem for nonlinear systems is to appeal to the results available in the linear case, that is, via linearization. This approach, however, generally only provides a local result, which has an attractive region of robustness locally.

Alternatively, LM networks formulate linear (affine) model to approximate the nonlinear systems instantaneously, and provide the possibility to design a non-local controller, which is conceptually simple and straightforward. Linear feedback control techniques can be utilized to stabilize the system. The following subsections present the stabilization of LLM systems by state feedback control first; then the results will be applied for the analysis and design of state feedback controllers for ALM systems.

4.3.1 Linear Local Controllers

When state feedback control is employed to design the controller, the main idea is to design each local state feedback controller so as to compensate each local model. For instance, each local controller has the same structure for each local model; the blending global controller has the same structure as the LM network. Consequently, the control signal is replaced with the feedback laws in the closed-loop system.

Assume each pair of (A_i, B_i) is controllable, design a matrix F_i to assign the eigenvalues of $A_i + B_i F_i$ to desired locations in the open left-half complex plane. Then the LC network based on the state feedback control approach can be written as follows:

$$u(t) = -\sum_{i}^{N} \rho_i F_i x(t) \tag{4.13}$$

Substituting (4.13) into (4.2), the closed-loop system can be described as

$$\dot{x} = \sum_{i=1}^{N} \sum_{j=1}^{N} \rho_i \rho_j \left(A_i - B_i F_j\right) x \tag{4.14}$$

Clearly, the origin is an equilibrium point of the closed-loop system. As an application of Theorem 1, Theorem 3 gives the sufficient conditions to guarantee the stability of system (4.14).

Theorem 3: The equilibrium point of system (4.14) is globally asymptotically stable if there exists a common positive definite matrix P such that

$$\left(A_i - B_i F_j\right)^T P + P\left(A_i - B_i F_j\right) < 0, \quad \text{for } \rho_i \rho_j \neq 0, \forall i, j = 1, \cdots, N \tag{4.15}$$

The controller design involves determining the local feedback gains F_i, $i = 1, \cdots, N$. The feedback gain F_i assures that $A_i + B_i F_i$ are Hurwitz, and that there is a positive definite matrix P satisfying Theorem 3. These requirements, on the one hand, satisfy a desired overall closed-loop system performance requirement; on the other hand, the global stability of the closed-loop system is guaranteed.

Wang et al. (1996) proposed the so-called parallel-distributed compensation (PDC) method as a design framework for the discrete time system and also modified Tanaka's stability theorem (Tanaka and Sugeno, 1992) to include a control algorithm. An important observation in Wang's approach is that the stability problem is structured as a standard feasibility problem with several LMIs when the feedback gains are predetermined, and can be solved numerically using an algorithm named the interior-point method. Thus the controller design is simple and natural. Basically, Wang's stability criterion is adopted and tailored here for the analysis of the stability problem of continuous time LLM systems.

Rewrite (4.14) as

$$\dot{x} = \sum_{i=1}^{N} \rho_i^2 (A_i - B_i F_i) x + 2 \sum_{i<j} \rho_i \rho_j G_{ij} x \tag{4.16}$$

where $G_{ij} = \dfrac{(A_i - B_i F_j) + (A_j - B_j F_i)}{2}, i < j$

Therefore theorem 3 may be reformulated as theorem 4 with relaxed constraints as follows:

Theorem 4: The equilibrium of system (4.14) is asymptotically stable in the large if there exists such a common positive definite P matrix that satisfies the following two conditions:

$$(A_i - B_i F_i)^T P + P(A_i - B_i F_i) < 0, \quad i = 1, 2, \cdots, N \tag{4.17}$$

$$G_{ij}^T P + P G_{ij} < 0\,,\ i < j \le N \qquad (4.18)$$

The controller design method based on this result is a trial-and-error method. Generally, after designing a local controller for each local model, the global stability conditions need to be checked. If the stability conditions are not satisfied, the design procedure needs to be repeated. As for the ALM blending systems, extra tools and analysis are required to obtain a closed-loop system with guaranteed stability, in addition to meeting the conditions in (4.17) and (4.18).

4.3.2 Affine local controllers

Assuming a set of state feedback gains F_i exist, which satisfy the conditions in theorem 4, the idea of canceling the offset term of the affine models is proposed here for controller design. The idea of directly canceling the offset term in the affine local model structure is not new. Hunt and Johansen (1997) utilized a feedforward loop to cancel the offset term in gain-scheduled LC networks. Such approach showed improvement in the transient performance of the closed-loop systems. The offset term in the blending systems plays an important role in the capture of the nonlinear dynamics. Introducing the offset term substantially improved the approximation accuracy of LM networks compared with the normal linear LM networks (McLoone, 2000, Fantuzzi and Rovatti, 1996). To best utilize the knowledge from the model, it is worthwhile to consider the offset term for controller design.

It is not hard to see that canceling the offset term $d(x,u)$ can be done by directly subtracting the control signal u in equation (4.13), as shown in equation (4.19). Therefore, using feedback control converts an affine local state equation into a controllable linear state equation by canceling the offset term locally. However, because of the interpolation technique involved, the closed-loop system is not generally a simple sum of linear subsystems. A detailed analysis is given below.

Design a set of feedback gains F_i such that each $A_i + B_i F_i$ is Hurwitz. The feedback control signal is given by

$$u = -\sum_{i=1}^{N} \rho_i (\sigma_i d_i + F_i x) \tag{4.19}$$

where $\sigma_i = [\sigma_1 \quad \cdots \quad \sigma_N]$. Substituting (4.19) into (4.2) results in the overall closed-loop system description

$$\dot{x} = \sum_{i=1}^{N}\sum_{j=1}^{N} \rho_i \rho_j (A_i - B_i F_j) x + \sum_{i=1}^{N}\sum_{j=1}^{N} \rho_i \rho_j (d_i - B_i \sigma_j d_j) \tag{4.20}$$

There are two parts in equation (4.20). The first is the same as (4.14) in section 4.3.1. It plays a vital role in stabilizing the global system; the second part is nonlinear and is parameter dependent (it relies on the states, input or output parameters applied in validity functions) due to the interpolation technique involved, but it is bounded. Under the assumption that the second part changes slowly, an integrator was proposed in chapter 3 for the controller design. Here, the second part of (4.20) will be taken as modelled nonlinear dynamics (related to the affine terms) in the controller design. Rewrite the second part of (4.20) as follows:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} \rho_i \rho_j (d_i - B_i \sigma_j d_j)$$

$$= \sum_{i=1}^{N} \rho_i^2 (d_i - B_i \sigma_i d_i) + 2\sum_{i<j} \rho_i \rho_j \theta_{ij} \tag{4.21}$$

where $\theta_{ij} = \dfrac{(d_i - B_i \sigma_j d_j) + (d_j - B_j \sigma_i d_i)}{2}, i < j$

Consider the special case that $B_i = B, i = 1, \cdots, N$ first. Then,

$$\theta_{ij} = \frac{(d_i - B\sigma_j d_j) + (d_j - B\sigma_i d_i)}{2} = \frac{(d_i - B\sigma_i d_i) + (d_j - B\sigma_j d_j)}{2},$$

If we set $d_i - B\sigma_i d_i = 0$, $i = 1, \cdots, N$, then $\theta_{ij} = 0$, thus the closed-loop system is associated only with the first part of (4.20). Therefore, the cancellation of the offset term leads to a linear system description. Consequently, the stabilization problem for the nonlinear system is reduced to a stabilization problem for a set of controllable linear systems. We can proceed to design a stabilizing linear state feedback controller to locate the eigenvalues of the closed-loop system in the open left half plane, as presented in section 4.3.1. However, setting $d_i - B\sigma_i d_i$ equal to zero is generally only possible when B is full rank or else matched to the offset d_i.

Consider the more general case, i.e. B_i does not have to be equal to B_j $\forall i \neq j$, then

$$\theta_{ij} = \frac{(d_i - B_i\sigma_j d_j) + (d_j - B_j\sigma_i d_i)}{2}$$

$$= \frac{(d_i - B_i\sigma_i d_i) + (d_j - B_j\sigma_j d_j) + (\sigma_i d_i - \sigma_j d_j)(B_i - B_j)}{2}$$

If we set $d_i - B_i\sigma_i d_i = 0$, $i = 1,\cdots,N$, then

$$\theta_{ij} = \frac{(\sigma_i d_i - \sigma_j d_j)(B_i - B_j)}{2} \tag{4.22}$$

θ_{ij} is proportional to the difference between the model parameters B_i, $i = 1,\cdots,N$ and a difference functionally related to the designed controller offset parameter σ_i, $i = 1,\cdots,N$. Thus the second part of (4.21) can be written as

$$2\sum_{i<j} \rho_i \rho_j \frac{(\sigma_i d_i - \sigma_j d_j)(B_i - B_j)}{2} \tag{4.23}$$

Now, $0 < \rho_i < 1$, $i = 1,\cdots,N$ is the weighting function of i^{th} local model, so (4.23) is bounded by

$$\delta = \frac{N \cdot (N-1)}{8} \max\left(\left\|(\sigma_i d_i - \sigma_j d_j)\cdot(B_i - B_j)\right\|\right) \tag{4.24}$$

for a finite number N of affine blending systems. Deducing (4.22) from (4.23) involves employing the identity $\frac{a+b}{2} \leq ab$ and the fact that the number of items summed in (4.23) is $1+2+\cdots+(N-1)=\frac{N\cdot(N-1)}{2}$.

We distinguish two cases:

1. Consider the case that $\rho_i = 1$ and $\rho_j = 0$, $i \neq j, i, j = 1,\cdots,N$.

2. Consider that the operating point resides in more than one region $\Re_i$, $i = 1,\cdots,N$, so that the local systems are interpolated. This is the general case.

For the special case 1, $\theta_{ij} = 0$, the design and analysis issue is simplified as discussed previously, i.e. when the system has a common B.

To deal with the general case, the closed-loop system can be written as

$$\dot{x} = \sum_{i=1}^{N}\sum_{j=1}^{N}\rho_i\rho_j\left(A_i - B_iF_j\right)x + 2\sum_{i<j}\rho_i\rho_j\left(\sigma_i d_i - \sigma_j d_j\right)\left(B_i - B_j\right) \tag{4.25}$$

System (4.25) includes an offset term, which is nonlinear parameter dependent.

Applying the stability analysis in section 4.2.2.2, Lemma 3 is developed below.

Lemma 3: The affine blending system (4.2) is uniformly bounded for all $t \geq t_0$ via state feedback controller (4.19), if there exits a positive definite matrix P, which satisfies (4.17) and (4.18). Let $V = x^T Px$ be a Lyapunov function of the system (4.2), for some positive $0 < \theta < 1$, then the solution of the system $x(t)$ satisfies

$$\|x(t)\| \leq k\exp[-r(t-t_0)]\|x(t_0)\|, \quad \forall t_0 \leq t < t_1$$

and $\|x(t)\| \leq b, \forall t \geq t_1$

for some finite time t_1, where $k = \sqrt{c_2/c_1}$, $r = \dfrac{(1-\theta)c_3}{2c_2}$, $b = \dfrac{2\lambda_{\max}(P)}{c_3}\sqrt{\dfrac{c_2}{c_1}}\dfrac{\delta}{\theta}$

Proof: Assuming $V = x^T P x$ is the Lyapunov function of the LLM system (4.14), i.e. there is a common positive definite P existing for each local system so that (4.17) and (4.18) are satisfied and

$$c_1\|x\|_2^2 \le V \le c_2\|x\|_2^2,\ c_1 = \lambda_{\min}(P), c_2 = \lambda_{\max}(P)$$

Then we define

$$(A_i - B_iF_i)^T P + P(A_i - B_iF_i) = -Q_{il},\ \text{for } i = 1,\cdots,N$$

$$\text{and} \qquad G_{ij}^T P + PG_{ij} = -Q_{ij},\ \text{for } i < j, i, j = 1,\cdots,N.$$

Based on Lemma 1 and lemma 2, the differential of the Lyapunov function can be written as

$$\dot{V} = x^T\left(P\sum_{i=1}^{N}\sum_{j=1}^{N}\rho_i\rho_j(A_i - B_iF_j) + \sum_{i=1}^{N}\sum_{j=1}^{N}\rho_i\rho_j(A_i - B_iF_j)^T P\right)x + 2x^T P\left(2\sum_{i<j}\rho_i\rho_j\theta_{ij}\right)$$

$$= x^T\left(\sum_{i=1}^{N}\rho_i^2\left[(A_i - B_iF_i)^T P + P(A_i - B_iF_i)\right]\right)x$$

$$+ 2x^T\sum_{i<j}\rho_i\rho_j(G_{ij}^T P + PG_{ij})x + 2x^T P\left(\sum_{i<j}\rho_i\rho_j(\sigma_i d_i - \sigma_j d_j)(B_i - B_j)\right)$$

$$= -x^T\left(\sum_{i=1}^{N}\rho_i^2 Q_{il} + 2\sum_{i<j}\rho_i\rho_j Q_{ij}\right)x + 2x^T P\left(\sum_{i<j}\rho_i\rho_j(\sigma_i d_i - \sigma_j d_j)(B_i - B_j)\right)$$

$$\le -\left(\min(\lambda\min(Q_{il})) + \frac{N\cdot(N-1)}{2}\cdot\min(\lambda_{\min}(Q_{ij}))\right)\|x\|_2^2 + 2\|x\|_2\lambda_{\max}(P)\delta$$

(From Lemma 1 and equation (4.23) and (4.24))

$$\leq -c_3\|x\|_2^2 + 2\|x\|_2 \lambda_{\max}(P)\delta$$

$$= -(1-\theta)c_3\|x\|_2^2 - (\theta c_3 - 2\lambda_{\max}(P)\delta)\|x\|_2$$

$$\leq -(1-\theta)c_3\|x\|_2^2, \ \forall \|x\|_2 \geq \frac{2\lambda_{\max}(P)\delta}{\theta c_3}$$

Apply theorem 2 to complete the proof. Lemma 3 shows the closed loop system is bounded by an ultimate bound *b*, which, in strong part, depends on the interpolation function ρ_i, $i = 1, \cdots, N$, which in turn depends on the states, system input and system output.

4.4 Concluding remarks

Blending systems with affine local models show improved modelling accuracy over normal linear model based blending systems. However, affine systems do not carry on the continuity with the well-established linear approaches. The blended offset term partially describes the dynamic properties of the overall systems and should be considered in the controller design. The offset terms in affine local models move the equilibria point of the system away from the origin x=0. The stability analysis is consequently difficult.

This chapter utilises the ultimate boundedness theory to the stability analysis of affine blending systems, in which the offset term is nonlinear, parameter dependent (on states, and input and output signals) and bounded. The approach considers the nonlinear offset term in affine blending systems as a non-vanishing 'perturbation' being added to the corresponding nominal linear blending systems.

Assuming the nominal linear blending system is exponentially asymptotically stable, i.e. there is a common positive definite matrix *P* existing for all the local systems, the corresponding affine blending system will be bounded by an ultimate value *b*, which is proportional to the maximum of the offset terms of the local systems. The smaller the bound b is, the smaller the deviation of the affine blending systems from the stabilizing origin of the linear blending systems.

A state feedback controller, with an extra term, is proposed to cancel the offset term of the affine blending systems. In some special cases, the overall compensated system is stable at the origin via the state feedback control; this happens only if the local systems have a common B, or if the weighting functions of the local models satisfy $\rho_i\rho_j = 0$, for $i \neq j,\ \ i, j = 1,\cdots,N$. Generally, the overall compensated system is bounded by an ultimate bound b, which is proportional to the differences between the local model parameter $B_i - B_j$, $i, j = 1,\cdots,N$, and the differences between the designed offset terms in the state feedback controller.

The contribution of this chapter might be limited currently. Future work will go on to give quantity analysis of the boundedness result related to the boundedness of the operating region.

In summary, the blending systems using affine local models, although widely applied in the literature, do not provide continuity with established linear methods. In terms of control, affine blending systems raise some difficult issues due to the varying nonlinear offset term, which blurs the transparency of representation of the global dynamics. Leith and Leithead (1998, 1999) propose a novel class of blended multiple model networks, namely, the velocity-based multiple model network, which establishes a rigorous, and direct relationship between the dynamic characteristics of a nonlinear system and those of a related family of linear systems. Since the representation of the linear members of the family can be combined to approximate the representation of the nonlinear system arbitrarily accurately, the family of linear systems is able to describe the entire dynamics of the nonlinear system. This is the topic of the next chapter.

Chapter 5

Velocity-Based Multiple Model Networks

5.1 Introduction

A static model gives information about the steady-state relationship between the input and the output signal. A dynamic model should give the relationship between the input and the output signal during transients. It is naturally much more difficult to capture dynamic behaviour. However, dynamic behaviour is very significant when discussing control problems. In an attempt to accurately model nonlinear dynamical systems, blended multiple-model systems have got increasing attention for their attractive continuity with well-established linear methods. Local model networks and fuzzy inference systems are some examples in which the local valid sub-models are easily interpreted; also, the weighted sum of the local sub-models provides a qualitative high-level description of the nonlinear system.

In chapter 2, the methods for structuring LM networks were introduced, in which the blending (interpolation) procedure is required to provide smooth interpolation, in some sense, between the local models, to achieve an accurate representation over the whole operating space with only a small number of local models. At any instant operating point, a linear representation of the process is formulated via combination of a set of linear (or affine) local models to approximate the dynamics of nonlinear systems. However, the blending procedure itself is inherently nonlinear, which raises questions about the ease of implementation of the approach. In chapter 4, the stability issue was investigated for the blending system. Although widely applied in the literature, it is noted that the offset (inhomogeneous) terms in affine blending systems are nonlinear and parameter dependent. They need to be carefully treated in the controller design in general.

Recent research (Leith and Leithead, 1998) queried the ease of interpretability of the multiple model frameworks for nonlinear systems and presented a novel class of blended multiple model networks, i.e. velocity-based (VB) multiple model networks, in which the global dynamics are directly related to the local models employed. The VB analysis and design framework associates a linearization with every operating point of a nonlinear system. The relationship between the nonlinear system and its VB linearisation is direct. Moreover, the underlying sub-models are continuous-time, velocity-based and linear, thus maintaining the continuity with existing linear techniques, which are well developed for analysis and controller design. Meanwhile, analytical results based on the application to a complex nonlinear continuous stirred tank reactor (CSTR) simulated process show that the velocity-based approach is ideally suited to the development of local controller (LC) networks (McLoone et al., 2001).

In chapter 3, a gain-scheduled LC network was applied to the control of the CSTR under the assumption that the CSTR system changes slowly. With regard to controller design, the VB approach does not inherently involve a slow variation requirement on the system (Leith and Leithead, 1999, 2001). However, although the VB approach shows significant advantages in representing the dynamics of the nonlinear systems, not many control applications have been developed based on VB multiple model networks, though there are some results reported (Leith and Leithead, 2001, Rong et al., 2001). This is partially due to some difficulties in practical implementation that engineers have to face; for example, the differential realization of the controller input signal and the 'drift' problem of steady-state errors.

McLoone and Irwin (2001) utilized sinusoids and constant signals to facilitate the formulation of differential signals in the simulation of modelling of the CSTR process and show that the steady-state errors could be accumulated as simulation proceeds. However, with regard to the controller design based on VB multiple model networks, so far not much work has been done.

This chapter proposes a novel controller design approach based on the VB multiple model networks via the application of integral controllers. This approach skilfully employs an integral action to eliminate the need for the derivative of the controller input. The solution locally incorporates an integrator in the state feedback loop. Meanwhile, a state observer is applied to adjust the 'drift' problem of steady-state errors in the feedback loop. A case study on the CSTR process highlights the feasibility and simplicity of the proposed approach in the application of VB multiple model networks to the control of complex nonlinear systems.

Furthermore, the literature review shows that a lot of work has been done regarding the conventional LM technique in both the continuous-time and discrete-time domains. However, all the studies relating to VB multiple model networks exist in the continuous-time domain. Considering that applications of digital computers are popular in the field of control and the potential capability of the VB multiple model network approach in the development of LC networks, this chapter develops and presents a discrete-time version of the VB multiple model representation. The modelling capabilities of the resulting nonlinear model are examined via simulation on the CSTR process.

The chapter is organised in the following sections. Section 5.2 briefly outlines the continuous-time VB multiple model network approach, while section 5.3 develops the state feedback integral controller based on the VB multiple model network. In section 5.4, the novel discrete-time version of the network is developed and simulation results are given for the CSTR process; the chapter ends with some conclusions and suggestions for future work in section 5.5.

5.2 Continuous Time VB Multiple Model Networks

5.2.1 VB Multiple Model Networks Formulation

Consider the general nonlinear state space SISO system (2.3), with state vector x and inputs u:

$$\dot{x}(t) = f(x(t), u(t))$$

$$y(t) = g(x(t), u(t)) \qquad (5.1)$$

Linearizing (5.1) about an operating point (x_0, u_0), ignoring the high order errors and keeping only the linear terms yields (2.6), which is rewritten as follows:

$$\delta\dot{x}(t) = f(x_0, u_0) + \frac{\partial f}{\partial x}\big|_{(x_0,u_0)} \partial x(t) + \frac{\partial f}{\partial u}\big|_{(x_0,u_0)} \partial u(t) \qquad (5.2)$$

where $\delta x(t) = x(t) - x_0$, $\delta u(t) = u(t) - u_0$. Notice that if the operating point (x_0, u_0) is an equilibrium point of the system, $f(x_0, u_0)$=0. This is the case for normal LM network models, although there are some contributions on off-equilibrium LM networks available (Johansen et al., 1998a). However, it is not necessary to linearize the system at the equilibrium point for the VB multiple model network, which allows linearization of the system at any instant operating point.

Define $A_0 = \frac{\partial f}{\partial x}\big|_{(x,u)=(x_0,u_0)}$, $B_0 = \frac{\partial f}{\partial u}\big|_{(x,u)=(x_0,u_0)}$, equation (5.2) can be rewritten as

$$\dot{x} = A_0 x + B_0 u + a_0 \qquad (5.3)$$

where $a_0 = f(x_0, u_0) - (A_0 x_0 + B_0 u_0)$. Differentiating (5.3) with respect to time gives the linear velocity-based system, whose state vector is denoted by $\breve{x}$, as follows.

$$\ddot{\breve{x}} = A_0 \dot{\breve{x}} + B_0 \dot{u} \qquad (5.4)$$

With the appropriate initial conditions, equations (5.2) and (5.4) give identical solutions, and therefore there is no approximation at this stage. Equation (5.4) establishes a direct relationship between the dynamics of the VB form of the nonlinear system and the VB linearisation near an operating point. Furthermore, members of the family of VB linearisation functions are all linear, which provides continuity with established linear theory and methods.

A velocity-based, blended, multiple-model system is formed by weighting several velocity-based linearized models as follows:

$$\ddot{\hat{x}} = \left(\sum_i A_i(\tilde{x}_i, u_i)\rho_i(\psi)\right)\dot{\hat{x}} + \left(\sum_i B_i(\tilde{x}_i, u_i)\rho_i(\psi)\right)\dot{u} \qquad (5.5)$$

where $A_i(\tilde{x}_i, u_i) = \partial f / \partial \tilde{x} \,|_{(\tilde{x},u)=(\tilde{x}_i,u_i)}$, $B_i(\tilde{x}_i, u_i) = \partial f / \partial u \,|_{(\tilde{x},u)=(\tilde{x}_i,u_i)}$ and $(\tilde{x}_i, u_i)$ is the freezing point of the i^{th} local model as below.

$$\ddot{\hat{x}} = A_i(\tilde{x}_i, u_i)\dot{\hat{x}} + B_i(\tilde{x}_i, u_i)\dot{u} \qquad (5.6)$$

in which $\dot{\hat{x}}$ denotes the state vector of the linearization function at that freezing point $(\tilde{x}_i, u_i)$.

The normalised weighting function is given by $\rho_i(\psi)$, where ψ is the scheduling vector as introduced in chapter 2. The dynamics of the blended system, about the operating point $(\tilde{x}_0, u_0)$ is now considered. The velocity-based linearized form of (5.5), at $(\tilde{x}_0, u_0)$, is simply obtained by freezing the validity function $\rho_i(\psi)$ at the operating point to produce the following linear system:

$$\ddot{\hat{x}} = \left(\sum_i A_i(\tilde{x}_i, u_i)\rho_i(\psi_0)\right)\dot{\hat{x}} + \left(\sum_i B_i(\tilde{x}_i, u_i)\rho_i(\psi_0)\right)\dot{u} \qquad (5.7)$$

With the appropriate initial conditions, the solution to (5.7) is initially tangential to the solution of the velocity-based multiple model system in (5.5). The dynamics of the multiple model system local to an arbitrary operating point are therefore the same as the dynamics of the corresponding frozen-form linear system at the same operating point. Rewriting (5.7) as

$$\ddot{\tilde{x}} = \sum_i \rho_i(\psi_0)\left(A_i(\tilde{x}_i,u_i)\dot{\tilde{x}} + B_i(\tilde{x}_i,u_i)\dot{u}\right) \qquad (5.8)$$

clearly highlights this direct relationship between the frozen-form (5.7) of the velocity-based blended system and the underlying local models (5.8) at $(\tilde{x}_0,u_0)$. Thus, at any arbitrary operating point, the global dynamics of the multiple model system are described by a straightforward weighted sum of the local model dynamics. Further detailed theoretical analysis of both conventional and velocity-based nonlinear representations can be found in (Leith and Leithead, 1999, McLoone et al., 2001, McLoone and Irwin, 2001).

5.2.2 Case Study

Recalling equation (5.5), we see that the input of the velocity-based multiple model is the time differential of the control signal u. The simulation diagram is shown in Figure 5-1. If the control signal u is step signal, then the input to the VB multiple model network would be an impulse. Practically, it is very difficult to formulate an impulse input signal because of the differential problem. This is an important issue that should be concerned.

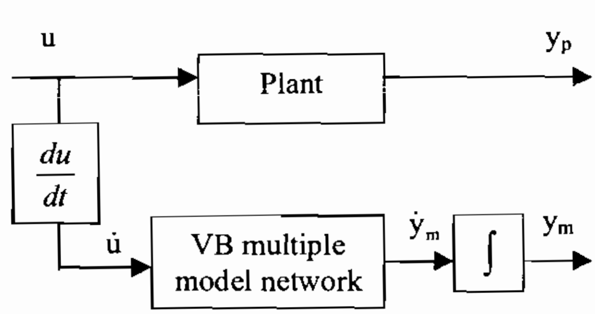


Figure 5-1 VB multiple model networks implementation

Mathematically, in the continuous time domain,

$$\dot{u}(t) = \lim_{\tau \to 0} \frac{u(t+\tau) - u(t)}{\tau}$$

For simplicity, in simulation, we employed the strategy in (McLoone and Irwin, 2001), in which sinusoids and constant signals were combined to facilitate the formulation of impulse signals. Two local models are employed to model the relationships between the coolant flow rate $q_c(t)$ and the product concentration, $C(t)$. These two local models are obtained by freezing the nonlinear velocity model at the appropriate linearization points:

$$C_o^1 = 0.062 mol/l, T_o^1 = 448.7522K, q_{co}^1 = 90.0 l/\min$$
$$C_o^2 = 0.1298 mol/l, T_o^1 = 432.9487K, q_{co}^1 = 110.0 l/\min$$

in which (C_o^i, T_o^i, q_{co}^i) denotes the linearization point of the i^{th} local model. The interpolation function is the same as used in chapter 2, in which $\sigma = [0.02, 0.03]$.

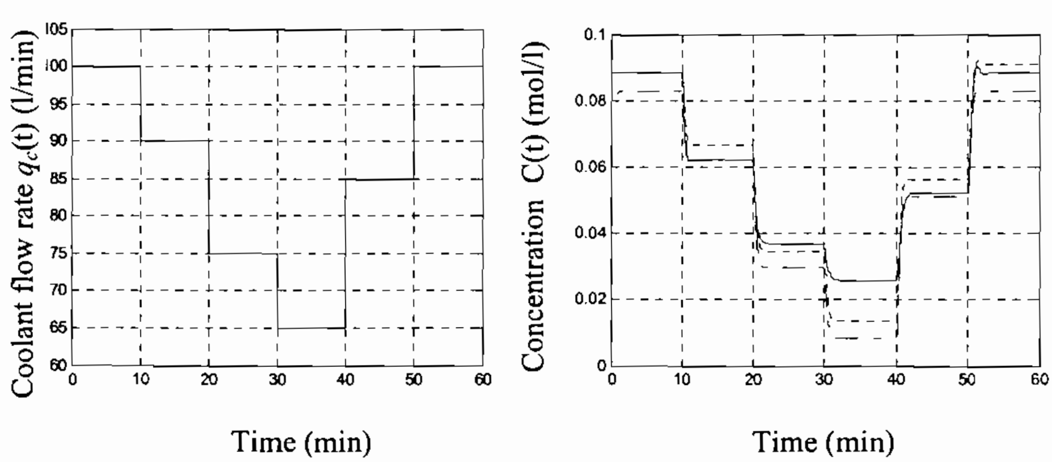
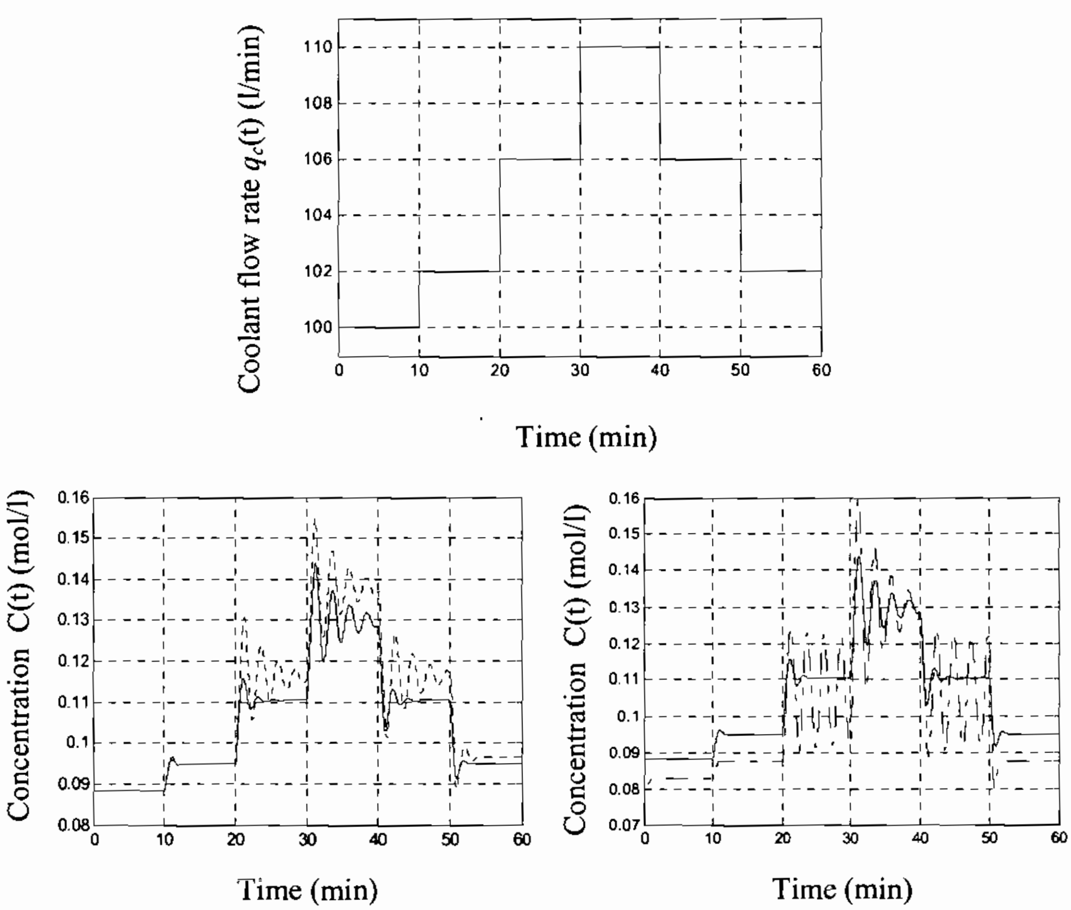


Figure 5-2 Comparison of output product concentration when $C(t)<0.1$

The solid lines represent the simulated CSTR process output, while the dotted line is from VB multiple model networks and the dash-dotted line is from the conventional LM networks.

Figure 5-2 and figure 5-3 show the coolant flow rate $q_c(t)$ step changes and comparison of the product concentration output *C(t)* from the CSTR process with the corresponding output from the velocity-based multiple model network and the conventional LM network in the continuous time domain. In Figure 5-2, coolant flow rate $q_c(t)$ varies from 65 l/min to 100 l/min. The CSTR lies in over-damped operating space ($C(t)<0.1$). Both of the modelling performances are poor in terms of the steady-state accuracy.



(a) VB multiple model network

(b) conventional LM network

Figure 5-3 Comparison of output product concentration when *C(t)*>0.1

The solid lines represent the CSTR process output, while the dotted line is from VB multiple model networks in (a) and dash-dotted line is from the conventional LM networks in (b).

However, in figure 5-3, coolant flow rate $q_c(t)$ varies from 100 l/min to 110 l/min. The CSTR lies in under-damped operating space ($C(t)$>0.1).the VB multiple model network shows much better dynamic capability than the conventional LM network especially, when the concentration output $C(t)$ goes to about 0.11-0.13 mol/l, where the CSTR is extremely under-damped. The conventional LM network fails to capture the dynamics of the CSTR. Meanwhile, the steady-state errors from VB multiple model networks are smaller than those from conventional LM network. This significant advantage of the VB multiple model network approach shows the potential of VB multiple model network applications in control. The next section will investigate the controller design issue based on VB multiple model networks.

More detailed analysis in the continuous time domain can also be found in (McLoone et al., 2001, Leith and Leithead 1998, 1999). Modelling properties of VB multiple model networks in the discrete time domain will be investigated in section 5.4.

5.3 VB Integral State Feedback Control Networks

Traditionally, integral actions can be employed in controller design to overcome the problem of steady-state errors. In many cases, it is difficult to obtain an accurate value for the plant gain, in part, because plants are typically nonlinear and the plant model is linearized at some particular point. Therefore, steady-state errors will result even though the model is sufficiently accurate for good feedback controller design. One solution is to incorporate an integral control term in the feedback loop.

Chapter 3 presented the application of integral actions in the LC network based on generalised predictive control approaches. While the affine terms in the local models cause difficulties for the controller design, the integrator is employed to evenly treat the affine terms the same as slow disturbance variations. Such an approach ignores the dynamics that those affine terms can describe. However, the VB local models are linear. This advantage makes it convenient to design local controllers based on well-developed control approaches for linear systems.

As presented in section 5.2, the VB multiple model networks instantaneously establish a direct linearisation near an operating point of the nonlinear systems. Therefore, the VB multiple model networks are able to capture the dynamics of nonlinear systems better than the conventional local model networks. This provides a potential to design a better controller based on the more accurate modelling capability of the dynamics of nonlinear systems. However, the input signal to the VB multiple model networks is the time differential of the control input signal as shown in Figure 5-1, which is difficult to formulate in practical application.

This section proposes a novel integral state feedback control network based on the VB multiple model networks, for the control of nonlinear systems. This method makes the controller design possible to have the best access of dynamic information available from the VB multiple model networks, which represent the better dynamics of the nonlinear system than the conventional local model networks do. Meanwhile, this method makes the implementation of the controller design feasible and practical based on VB multiple model networks. Therefore, the integral state feedback control network is able to allow better performance by utilising the dynamics of the nonlinear systems that the VB multiple model network can capture.

5.3.1 State Feedback Integral controller

As shown in section 5.2.1, each local model in velocity-based multiple model networks is linear. Therefore, for each VB local model, a state feedback integral controller can be simply designed for the purpose of achieving satisfactory dynamic response in terms of rise-time, overshoot, settling time or other measures of transient response.

This subsection will present the state feedback integral controller design approach based on the VB multiple model networks. For each local model, the state feedback integral controller is able to exactly match the closed loop characteristic equation of a feedback system to its desired value. This is often used in controller design where the performance criterion for the control system can be expressed in the classical control terms of transient response or frequency response.

A general structure of a state feedback integral controller is shown in Figure 5-4.

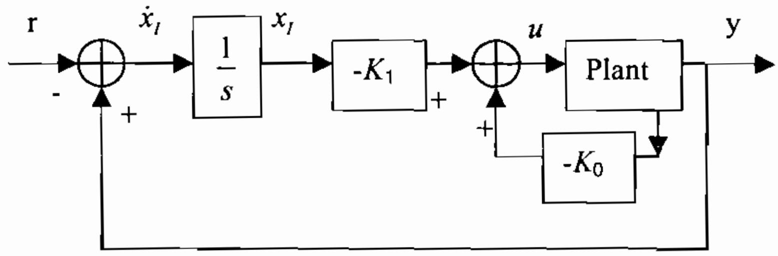


Figure 5-4 Conventional state feedback integral controller

Given a linear single-in-single-out (SISO) system description:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{F}\boldsymbol{x} + \boldsymbol{G}u \\ y &= \boldsymbol{H}\boldsymbol{x} \end{aligned} \tag{5.9}$$

Define $\dot{x}_I = y - r$, and then extend equation (5.9) as follows:

$$\begin{aligned} \begin{bmatrix} \dot{x}_I \\ \dot{\boldsymbol{x}} \end{bmatrix} &= \begin{bmatrix} 0 & \boldsymbol{H} \\ 0 & \boldsymbol{F} \end{bmatrix} \begin{bmatrix} x_I \\ \boldsymbol{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{G} \end{bmatrix} u - \begin{bmatrix} 1 \\ 0 \end{bmatrix} r \\ y &= \begin{bmatrix} 0 & \boldsymbol{H} \end{bmatrix} \begin{bmatrix} x_I \\ \boldsymbol{x} \end{bmatrix} \end{aligned} \tag{5.10}$$

Defining u=-[K_1 $\boldsymbol{K}_0$]$\begin{bmatrix} x_I \\ \boldsymbol{x} \end{bmatrix}$ and substituting it into (5.10) gives

$$\begin{bmatrix} \dot{x}_I \\ \dot{\boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} 0 & \boldsymbol{H} \\ -\boldsymbol{G}K_1 & \boldsymbol{F} - \boldsymbol{G}\boldsymbol{K}_0 \end{bmatrix} \begin{bmatrix} x_I \\ \boldsymbol{x} \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} r \tag{5.11}$$

We set the eigenvalues of the extended matrix $\begin{bmatrix} 0 & \mathbf{H} \\ -\mathbf{G}K_1 & \mathbf{F} - \mathbf{G}\mathbf{K}_0 \end{bmatrix}$ to the desired poles, and then K_1 and $\boldsymbol{K}_0$ can be calculated.

Recall the velocity-based linearization local model (5.6), for the i^{th} VB local model, equation (5.6) could be rewritten as equation (5.12), which defines $\overline{\boldsymbol{w}} = \dot{\boldsymbol{x}}$ and $\dot{\overline{w}}_I = \dot{y} - \dot{r}$.

$$\dot{\overline{w}} = A_i\overline{w} + B_i\dot{u}$$
$$\dot{y} = C\overline{w} \qquad (5.12)$$

Let $\dot{u} = -[K_1 \quad K_0]\begin{bmatrix} \overline{w}_I \\ \overline{w} \end{bmatrix}$, an integral controller can be designed to satisfy some specifically assigned requirement using the above method. Then a little change can be made from Figure 5-4 to Figure 5-5.

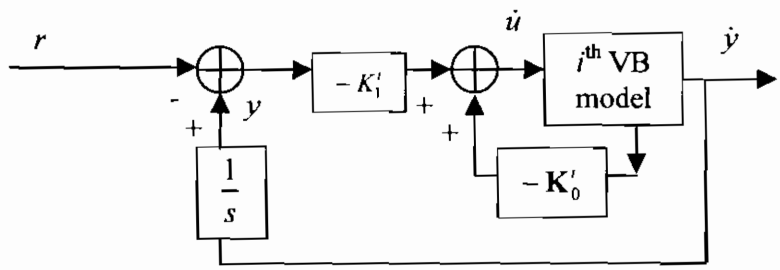


Figure 5-5 The i^{th} local VB integral controller

As for the blending VB multiple model networks, we define $w = \dot{x}$. Considering the single-input-single-output (SISO) system, equation (5.7) is rewritten as

$$\dot{w} = \sum_i \rho_i A_i w + \sum_i \rho_i B_i \dot{u} \qquad (5.13)$$

It should be noted that the local VB subsystems share common states, as does the gain-scheduled LM controller network introduced in chapter 3. The VB multiple model networks provides a global nonlinear system representation from a set of linear models together with an interpolation function. The controlled closed-loop system is described by

$$\begin{bmatrix} \dot{w}_I \\ \dot{w} \end{bmatrix} = \begin{bmatrix} 0 & C \\ -\sum_i\sum_j \rho_i\rho_j B_i K_1^j & \sum_i \rho_i A_i - \sum_i\sum_j \rho_i\rho_j B_i K_0^j \end{bmatrix} \begin{bmatrix} w_I \\ w \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \dot{r} \qquad (5.14)$$

Then, the characteristic matrix of (5.14) is

$$\begin{bmatrix} 0 & \boldsymbol{C} \\ -\sum_i\sum_j \rho_i\rho_j \boldsymbol{B}_i K_1^j & \sum_i \rho_i \boldsymbol{A}_i - \sum_i\sum_j \rho_i\rho_j \boldsymbol{B}_i \boldsymbol{K}_0^j \end{bmatrix}$$

$$= \sum_i\sum_j \rho_i\rho_j \begin{bmatrix} 0 & \boldsymbol{C} \\ \boldsymbol{B}_i K_1^j & \boldsymbol{A}_i - \boldsymbol{B}_i \boldsymbol{K}_0^j \end{bmatrix}$$

$$= \sum_i \rho_i^2 \begin{bmatrix} 0 & \boldsymbol{C} \\ \boldsymbol{B}_i K_1^i & \boldsymbol{A}_i - \boldsymbol{B}_i \boldsymbol{K}_0^i \end{bmatrix} + 2\sum_{i<j} \begin{bmatrix} 0 & \boldsymbol{C} \\ \dfrac{\boldsymbol{B}_i K_1^j + \boldsymbol{B}_j K_1^i}{2} & \dfrac{\left(\boldsymbol{A}_i - \boldsymbol{B}_i \boldsymbol{K}_0^j\right) + \left(\boldsymbol{A}_j - \boldsymbol{B}_j \boldsymbol{K}_0^i\right)}{2} \end{bmatrix}$$

Employing the stability theorems discussed in chapter 4 and comparing the above equation with equation (4.17), the stability of system (5.14) can be investigated in the same way as described in Theorem 4 (chapter 4). If there is a common positive P existing for each local feedback system, the overall closed-loop system is stable and it stabilizes to the origin. This shows another advantage of the VB approach over the conventional affine LM network, which is a bounded system generally. The overall closed-loop system is shown in Figure 5-5.

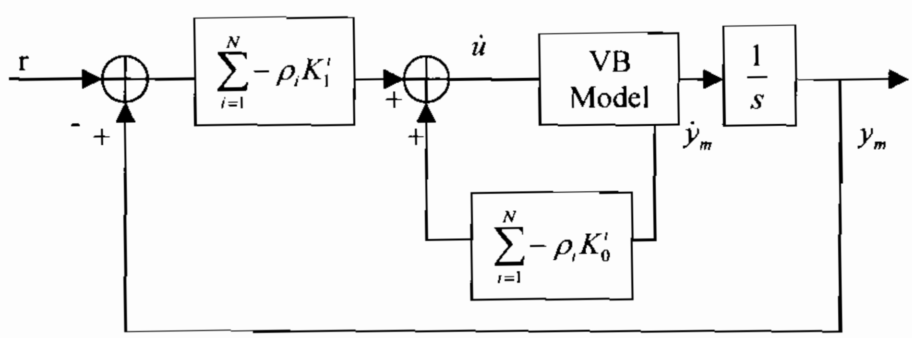


Figure 5-6 VB integral state feedback control network

When the VB integral state feedback controller network is applied to the plant, a framework is proposed as shown in Figure 5-7, which introduces an observer to the modelling loop. As mentioned in section 5.2.2, the velocity-based multiple model network has a weakness in static modelling accuracy, for there are steady-state errors existing, although the VB approach shows an attractive capability in capturing the dynamics of nonlinear systems. Moreover, these steady-state errors accumulate as the simulation continues and the VB multiple model outputs drift away from the nonlinear system output (McLoone et al. 2001). So, it is desirable to bring the VB multiple model outputs back to the proper operating point when we design a model-based controller as shown in Figure 5-7.

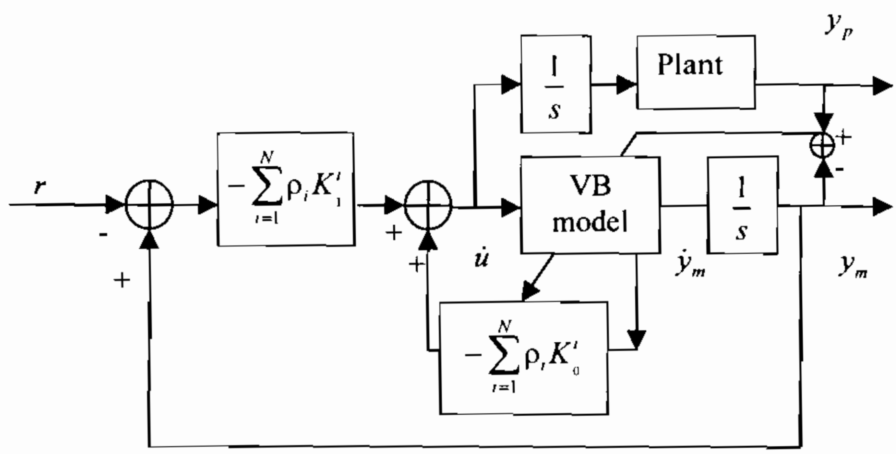


Figure 5-7 VB integral state feedback control network application to a plant

5.3.2 Case Study

The process continuous stirred tank reactor (CSTR) introduced in chapter 2 is considered as a case study. The CSTR is a highly nonlinear process. The objective of the controller design is to control the concentration output *C(t)* by manipulating temperature output *T(t)*, which is carried out by adjusting the effluent flow rate $q_c(t)$.

The VB multiple model network used here consists of two nominated velocity-based local models as shown in section 5.2.2. The two local controllers, corresponding to the velocity-based local models linearized at the nominal operating points ($C_o^1 = 0.062 mol/l$, $T_o^1 = 448.7522K$, $q_{co}^1 = 90.0 l/\min$ and $C_o^2 = 0.1298 mol/l$, $T_o^1 = 432.9487K$, $q_{co}^1 = 110.0 l/\min$ respectively), can be arbitrarily designed to give a desired closed loop characteristic form with a dominant set of closed-loop poles, by applying the controller structure proposed in section 5.3.1.

In the case study, two integral state feedback control networks (IC1 and IC2) have been designed. In IC1, each local controller is designed to give a settling time of 0.5 minutes; In IC2, each local controller is designed to give a settling time of 0.3 minutes. The reason for choosing the settling time of 0.5 minutes is for the purpose to give a fair comparison with the gain-scheduled LC networks described in chapter 3. IC2 with the settling time of 0.3 minutes facilitates the showing of the impact of local control design to the global performance based on the proposed VB integral state feedback control networks by the comparison with IC1.

A system whose poles are the roots of a Bessel polynomial is known to possess a desirable servo response by choosing the closed-loop pole locations (Vaccaro, 1995). Based on the Bessel polynomial, a third order system with 0.5 minutes settling time should have their poles set as $-7.9336 \pm 7.5690i$ and -10.0186; a third order system with 0.3 minutes settling time should have their poles set as $-13.2227 \pm 12.6150i$ and –16.6977. Their corresponding controller gains are calculated according to section 5.3.1, as follows:

IC1: $K_1^1 = 2.6285e4$; $K_1^2 = 3.1679e4$

$$K_0^1 = [-17.4651 \quad -30.9764];\ K_0^2 = [-4.162e2 \quad 2.0227e3]$$

IC2: $K_1^1 = 1.2169e5$; $K_1^2 = 1.4666e5$

$$K_0^1 = [-35.0047 \quad -52.0459];\ K_0^2 = [4.3450e3 \quad 1.1581e4]$$

where K_j^i denotes the j^{th} gain for the i^{th} local model. To examine the performance of the proposed integral controller design approach, a set of simulation experiments have been designed.

- Experiment E1 entails a comparison of the servo responses of two VB integral control networks IC1 and IC2.

As in chapter 3, two sets of step changes are designed in the relatively over-damped operating regime ($C(t)$<0.1) and in the relatively under-damped operating regime ($C(t)$>0.1). The simulation results are shown in Figure 5-8 and 5-9 respectively.

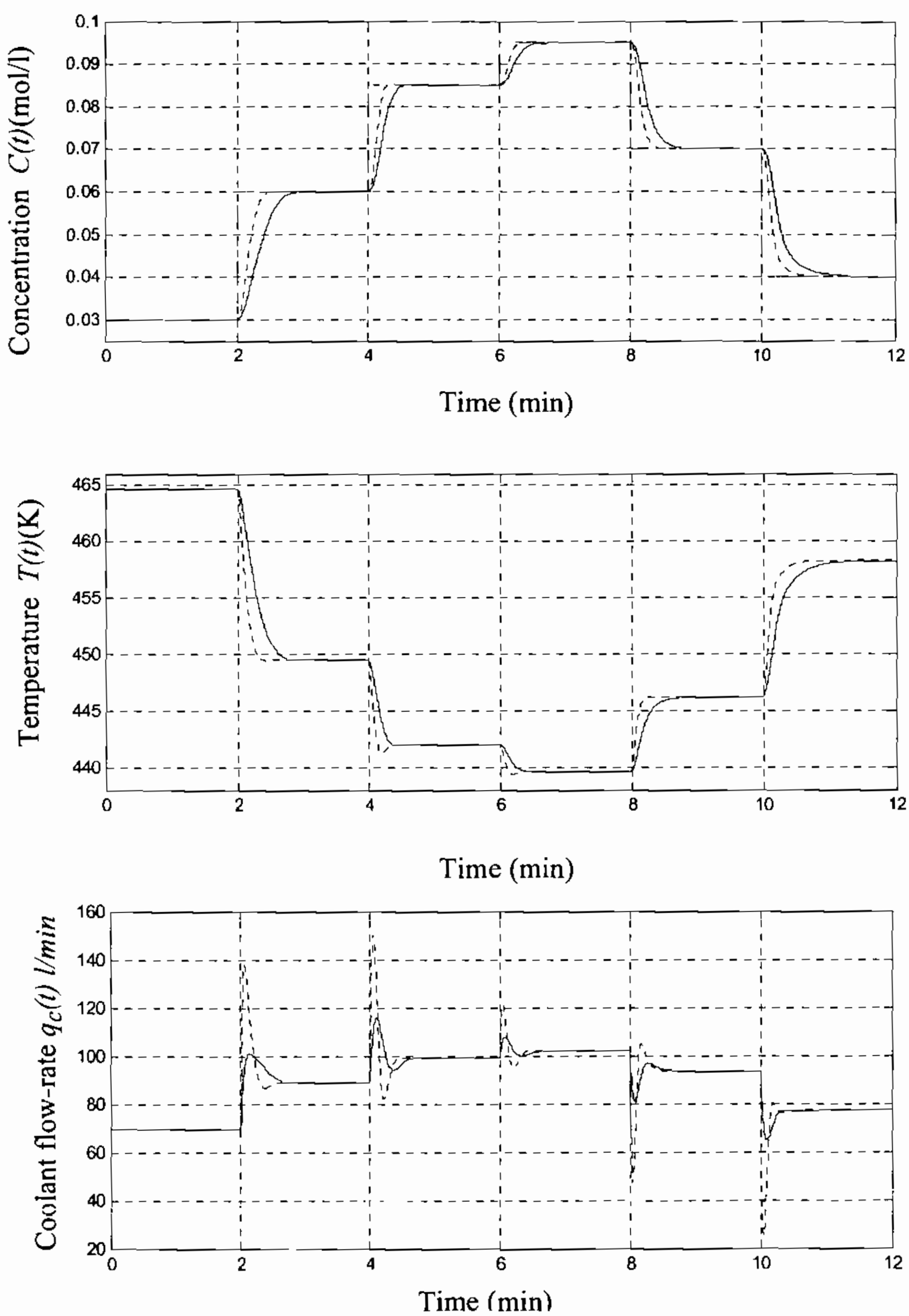


Figure 5-8 Closed-loop integral feedback controlled CSTR output for over-damped operating regimes

The dash-dotted line is the set point; the solid line is from IC1 (settling time 0.5 minutes); the dotted line is from IC2 (settling time 0.3 minutes).

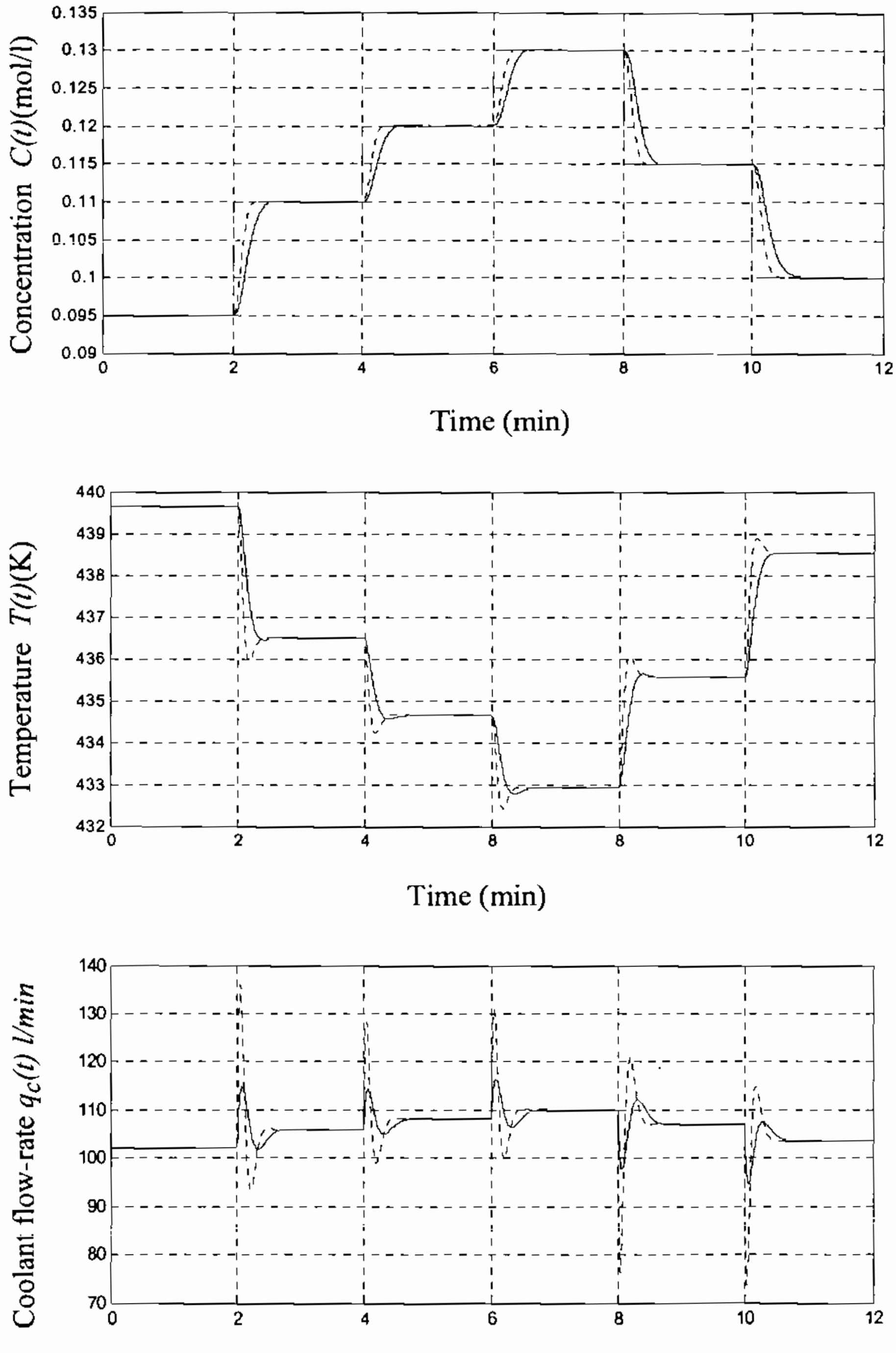


Figure 5-9 Closed-loop integral feedback controlled CSTR output for over-damped operating regimes

The dash-dotted line is the set point; the solid line is from IC1 (settling time 0.5 minutes); the dotted line is from IC2 (settling time 0.3 minutes)

In both of the figures, the performance of the product concentration output *C(t)* and the temperature output *T(t)* is satisfactory from both IC1 and IC2. Meanwhile, the product concentration output and the temperature output from IC2 always track the step change faster and settle down faster than those from the IC1 as expected. Moreover, the control inputs, i.e. the coolant flow rates $q_C(t)$, from IC2 shows more aggressive and quicker response than those from IC1 in both figure 5-8 and 5-9.

One thing that should be emphasized here is to differentiate the desired local control target (for the local system) from the expected global performance. The design approach described sets the desired control target locally only, but no desired global control target is set directly. For each step change from IC1, it normally settles down in 0.5 minutes, although it sometimes takes a bit longer especially during over-damped area; for each step change from IC2, it is normally able to settle down in 0.3 minutes. Although it takes a bit longer than 0.3 minutes, but it never takes longer than IC2 to settle down. It is clear from the simulation that the better and the more rigorous the local controller is designed, the better performance could be expected for the global system.

- Experiment E2 entails a comparison of the disturbance rejection of the two VB integral state feedback control networks IC1 and IC2.

A set of step changes has been designed to cover the whole operating regime. It starts from 0.03mol/l, with 0.03 mol/l intervals in over-damped area and with 0.01 mol/l in highly dynamic area, i.e. the under-damped area, as shown in Figure 5-10. The simulation results are shown in Figure 5-11. The disturbance introduced to the system is a set of impulses with 30 *l/min* coolant flow rate as shown in Figure 5-11.

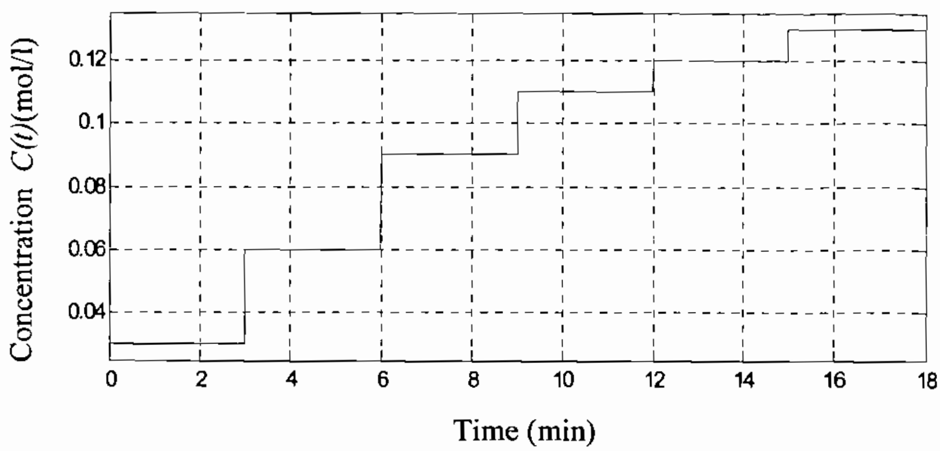
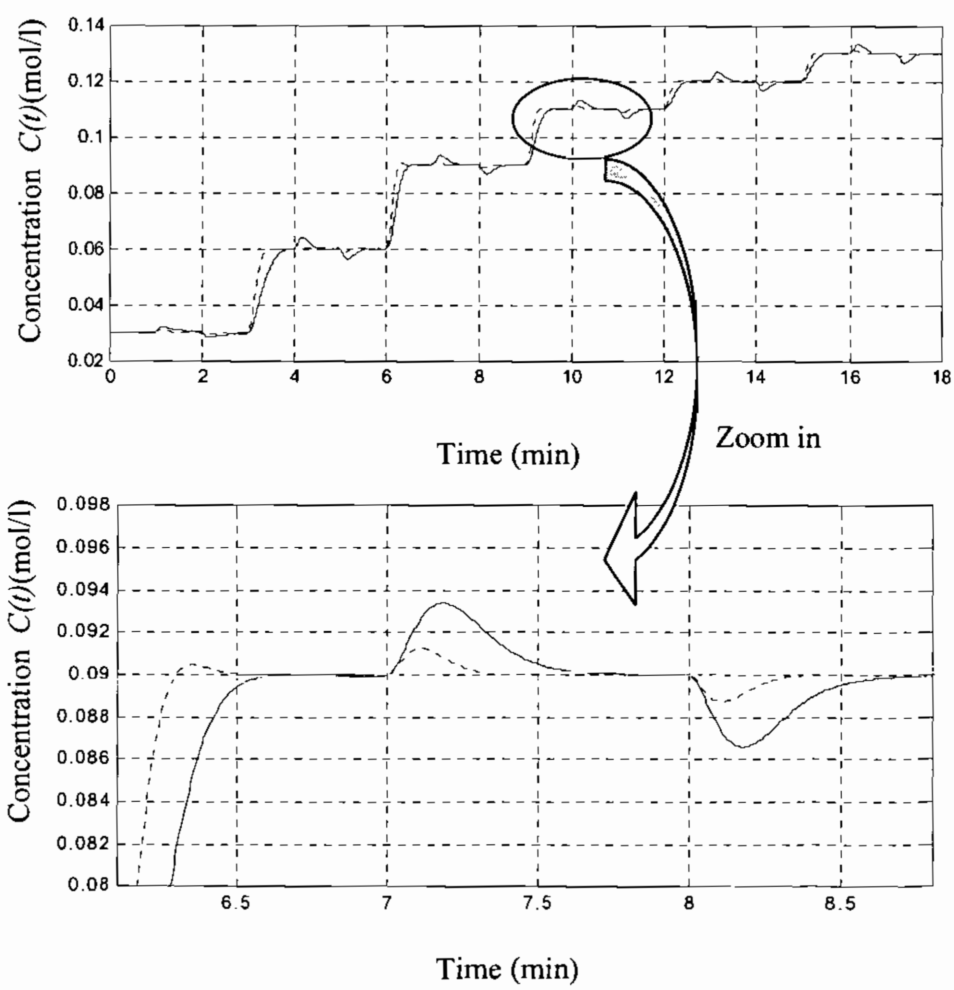


Figure 5-10 Step changes for disturbance rejection simulation



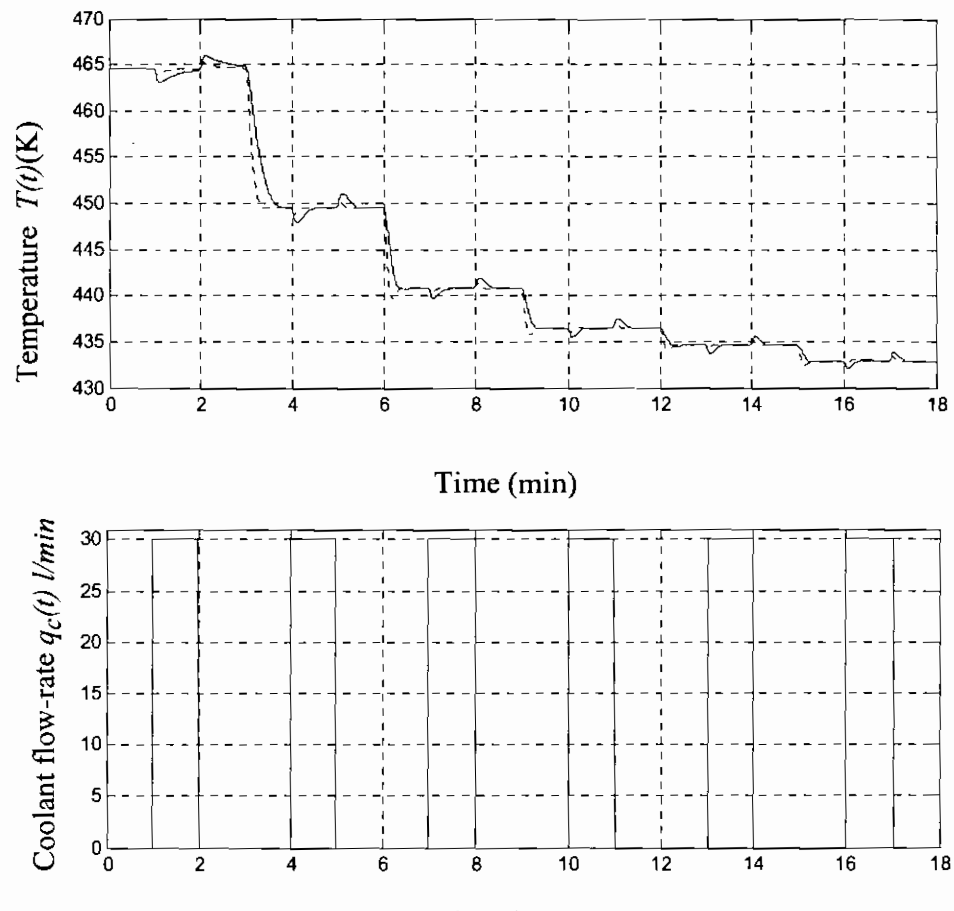
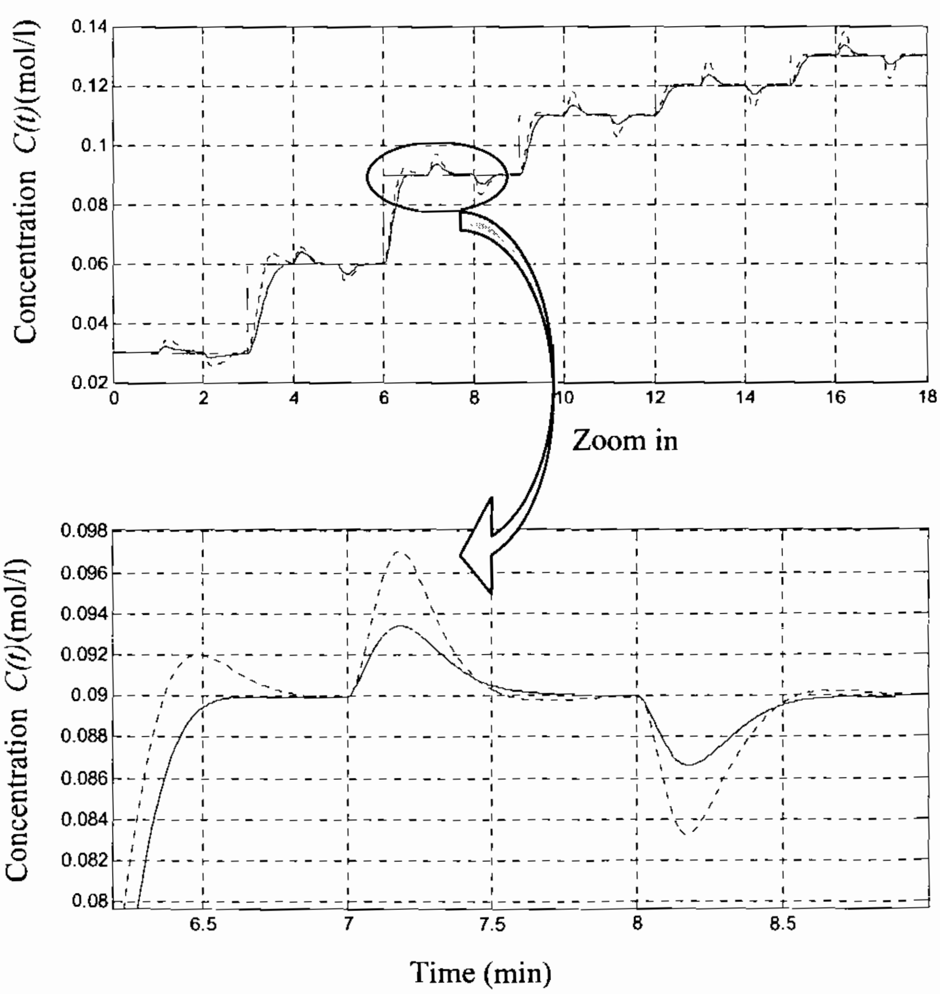


Figure 5-11 Comparison of disturbance rejection performance from IC1 and IC2
Solid line is from IC1 and dotted line is from IC2

Both IC1 and IC2 show good disturbance rejection ability in concentration output and temperature output, although IC2 shows even stronger robustness than IC1 does. The concentration output from both IC1 and IC2 goes back to the set point quickly after a tiny vibration when the disturbance change occurs. Meanwhile, IC2 shows much better robustness than IC1. IC1 shows only half of the overshoot that the IC1 gives, corresponding to the disturbance, as shown in the zoom-in figure. It is clear that the simulation shows that the better and the rigour the local controller are designed, the better and the more robust the global performance intends to be. The strong disturbance rejection performance shows an attractive contribution of the proposed integral state feedback control approach based on the VB multiple model networks. To firmly show this contribution, experiment E3 is designed.

- Experiment E3 entails a comparison of the disturbance rejection performance from the VB integral state feedback control networks IC1 and that from the gain-scheduled LC network with two local controllers (C1) proposed in chapter 3, which employed generalised predictive controllers as the basic local controllers.

The set of step changes has been designed for experiment E2, as shown in Figure 5-10, are used in this experiment E3 for the purpose of fair comparison. The main advantage of using multiple model networks is to obtain good performance globally rather than at certain local operating regime only, so it is important to show simulation results all over the operating regime rather than under certain limited operating conditions. Simulation results are shown in Figure 5-12.



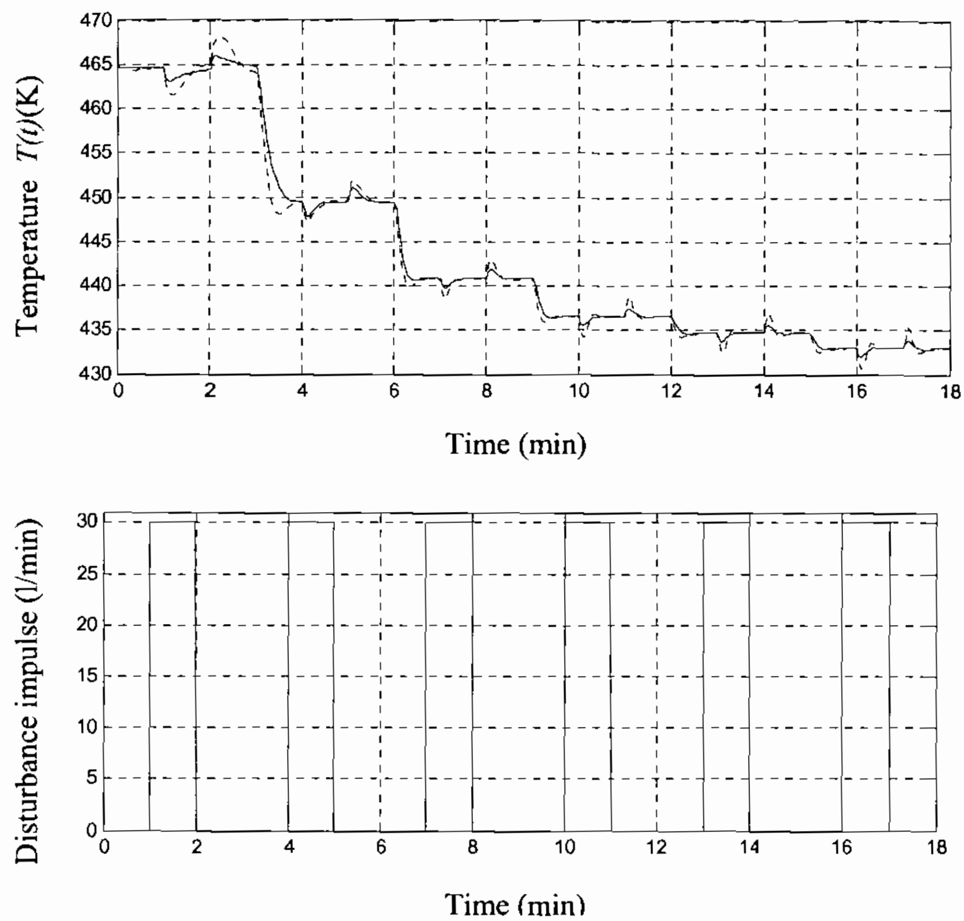


Figure 5-12 Comparison of disturbance rejection from IC1 and C1

The solid line is from the VB integral feedback network (IC1) controlled system and the dashed line is from the gain-scheduled LC network (C1) controlled system.

Obviously, the integral controller based on VB multiple model networks IC1 allows much better regulator performance compared with the gain-scheduled local controller networks C1, especially when $C(t)$ is equal to or greater than $0.09 mol/l$, i.e. in the relatively under-damped operating area, the overshoot of the results from the IC1 is less than half of those from C1 when disturbance occurs, as shown in the zoomed figure in Figure 5-12. This result also, in part, demonstrates the advantage of the velocity-based linearization, which applies not only at the equilibrium point but at any operating point, in capturing the dynamics instantly, so that the instant change of system dynamics can be fed back to the closed-loop and be properly controlled.

5.3.3 Comments

State feedback controllers have been widely applied in control engineering. The novel element of the proposed approach is that it introduces a combination of integral action and the state feedback control approach for the controller design based on the VB multiple model networks. It skilfully utilizes the integral action to eliminate the differential action necessary for the input signal; such differential action is normally not feasible in a practical implementation but required for the velocity-based approach.

This approach doesn't involve time-consuming numerical optimisation methods, like local controller (LC) network approach based on local model (LM) networks. It shows simplicity in controller design and continuity with the well-established linear control methods and theories for controller design and analysis.

Moreover, the proposed integral controller design approach based on VB multiple model networks shows excellent servo and regulator performance in simulation. This promising advantage is convincing when evaluating the potential of the application of VB multiple model networks in the control of complex dynamic nonlinear systems.

5.4 Discrete Time VB Multiple Model Networks

Velocity-based approaches show potential in modelling and control for complex dynamic nonlinear systems. However, the literature review shows that all the studies relating to VB multiple model networks exist in the continuous-time domain. Considering the popular application of digital computers in the field of control and the inherent capability of VB multiple model networks for controller design, this section formulates the z-transform of the velocity-based multiple model networks (not limited to SISO process). It starts from the z-transform of the conventional LM networks; next, the discrete-time velocity-based multiple model networks are developed mathematically; then simulation results are shown by application to the continuous stirred tank reactor (CSTR) simulated process.

5.4.1 Discrete Time Conventional LM Networks

5.4.1.1 ZOH Equivalent Model Development

The continuous time input to the plant is a zero-order hold (ZOH) of the compensator output

$$\boldsymbol{u}(t)=\boldsymbol{u}(k),\ kT \le t < kT+T \qquad (5.15)$$

and the output of the plant is sampled by an A/D converter:

$$y(k)=y(kT) \qquad (5.16)$$

Assume that we have a state space model (***A, B, C, D***) for the plant G(s); that is, the behaviour of the plant is governed by the following equations:

$$\begin{aligned}\dot{\boldsymbol{x}}(t)&=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t)\\ y(t)&=\boldsymbol{C}\boldsymbol{x}(t)+\boldsymbol{D}\boldsymbol{u}(t)\end{aligned} \qquad (5.17)$$

Because (5.17) is a first order differential equation, if the value of $\boldsymbol{x}(t)$ is known at some time t_0, then the value of $\boldsymbol{x}(t)$ at future times is given by

$$\boldsymbol{x}(t)=e^{A(t-t_0)}\boldsymbol{x}(t_0)+\int_{t_0}^{t} e^{A(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau \qquad (5.18)$$

where the symbol e^{At} stands for the matrix exponential function. If $t_0=kT$ and $t=kT+T$, where T is the sampling time, then (5.18) gives an update formula for the state vector at sampling instants. That is, integrating the state equation over one sample period yields

$$\boldsymbol{x}(kT+T)=e^{A(kT+T-kT)}\boldsymbol{x}(kT)+\int_{kT}^{kT+T} e^{A(kT+T-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau \qquad (5.19)$$

Now recall from (5.15) that in the interval of integration, the function $\boldsymbol{u}(t)$ is equal to $\boldsymbol{u}(k)$, a constant. This constant can be taken outside of the integral as follows:

$$x(kT+T)=e^{AT}x(kT)+\left[\int_{kT}^{kT+T}e^{A(kT+T-\tau)}\boldsymbol{B}d\tau\right]\boldsymbol{u}(k) \qquad (5.20)$$

This formula computes the value of the state vector $x(t)$ only at sampling instants $t=kT$. Thus, if we define a discrete time state space equation

$$x(k)=x(kT),$$

$$\boldsymbol{\Phi}=e^{AT},$$

$$\boldsymbol{\Gamma}=\int_{kT}^{kT+T}e^{A(kT+T-\tau)}\boldsymbol{B}d\tau \qquad (5.21)$$

Then (5.20) becomes the discrete time state space sequence:

$$x(k+1)=\boldsymbol{\Phi}x(k)+\boldsymbol{\Gamma}u(k) \qquad (5.22)$$

Note that $\boldsymbol{\Gamma}$ in (5.22) is a constant vector. Also, using the output equation of (5.17), we can write

$$y(k)=\boldsymbol{C}x(k)+\boldsymbol{D}u(k) \qquad (5.23)$$

Equations (5.22) and (5.23) constitute a discrete time system whose output, by construction, exactly matches the output of the analog system if its input is piecewise constant. Note that if $G(s)$ is linear and time invariant, then its ZOH equivalent will also be linear and time invariant.

5.4.1.2 Conventional LM Network Development

Assuming we have a set of linearized local models, each of which is governed by equation (5.2), for a nonlinear system described by equation (5.1), the local model is rewritten as follows:

$$\begin{aligned}\delta\dot{x}(t)&=\boldsymbol{A}_i(x_i,u_i)\delta x(t)+\boldsymbol{B}_i(x_i,u_i)\delta u(t)\\ y(t)&=\boldsymbol{C}x(t)\end{aligned} \qquad (5.24)$$

where $\delta\mathbf{x}(t)=\mathbf{x}(t)-\mathbf{x}_{ie}$, $\delta\mathbf{u}(t)=\mathbf{u}(t)-\mathbf{u}_{ie}$, $\mathbf{x}_{ie}$ and $\mathbf{u}_{ie}$ are the state vector and the input at the equilibrium points, near which the nonlinear system is linearized.

According to section 5.4.1.1, we have the ZOH equivalent models for each linearized model, as follows:

$$\delta\bar{x}(k+1)=\boldsymbol{\Phi}_i(x_i,u_i)\delta\bar{x}(k)+\boldsymbol{\Gamma}_i(x_i,u_i)\delta u(k)$$

$$y(k)=C\bar{x}(k) \tag{5.25}$$

in which $\delta\bar{x}(k)=\bar{x}(k)-x_{ie}$, $\delta\mathbf{u}(k)=\mathbf{u}(k)-\mathbf{u}_{ie}$, $\boldsymbol{\Gamma}_i=\int_{kT}^{kT+T}e^{A_i(kT+T-\tau)}\boldsymbol{B}_i d\tau$ and $\boldsymbol{\Phi}_i=e^{A_iT}$.

We can rewrite (5.25) as

$$\bar{x}(k+1)=\boldsymbol{\Phi}_i(x_i,u_i)(\bar{x}(k)-x_{ie})+\boldsymbol{\Gamma}_i(x_i,u_i)(u(k)-u_{ie})+x_{ie}$$

$$y(k)=C\bar{x}(k) \tag{5.26}$$

A normal, blended local model network system in the discrete time domain is formulated by weighing several local models:

$$x(k+1)=\sum_i\rho_i(\tilde{\psi}_0)(\boldsymbol{\Phi}_i(x_i,u_i)(x(k)-x_{ie})+\boldsymbol{\Gamma}_i(x_i,u_i)(u(k)-u_{ie})+x_{ie})$$

$$y(k)=Cx(k) \tag{5.27}$$

5.4.2 VB Multiple Model Network Development

Recalling the linearized i^{th} velocity-based local model in (5.6), we define $w=\dot{\bar{x}}$. Rewriting the equation as follows:

$$\begin{bmatrix}\dot{\bar{x}}\\ \dot{\bar{w}}\end{bmatrix}=\begin{bmatrix}0 & I\\ 0 & A_i(\tilde{x}_i,u_i)\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{w}\end{bmatrix}+\begin{bmatrix}0\\ B_i(\tilde{x}_i,u_i)\end{bmatrix}\dot{u} \tag{5.28}$$

Then, the linearized model output is

$$\bar{y} = \begin{bmatrix} c & 0 \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{w} \end{bmatrix} \tag{5.29}$$

For simplicity, we write (5.28) as follows

$$\dot{\overline{W}} = \breve{A}_i(\tilde{x}_i, u_i)\overline{W} + \breve{B}_i(\tilde{x}_i, u_i)\dot{u}$$

$$\bar{y} = \breve{C}\overline{W} \tag{5.30}$$

in which, $\breve{A}_i(\tilde{x}_i, u_i) = \begin{bmatrix} 0 & I \\ 0 & A_i(\tilde{x}_i, u_i) \end{bmatrix}$, $\breve{B}_i(\tilde{x}_i, \breve{u}) = \begin{bmatrix} 0 \\ B_i(\tilde{x}_i, u_i) \end{bmatrix}$, $\breve{C} = \begin{bmatrix} c & 0 \end{bmatrix}$, and $\overline{W} = \begin{bmatrix} \bar{x} \\ \bar{w} \end{bmatrix}$.

Then based on section 5.4.1.1, we have the velocity based local state-space model

$$\overline{W}(k+1) = \boldsymbol{\Phi}_i(\tilde{x}_i, u_i)\overline{W}(k) + \boldsymbol{\Gamma}_i(\tilde{x}_i, u_i)u(k)$$

$$\bar{y}(k) = C\overline{W}(k) \tag{5.31}$$

Where $\boldsymbol{\Phi}_i(\tilde{x}_i, u_i) = e^{\breve{A}_i(\tilde{x}_i, u_i)T}$, $\boldsymbol{\Gamma}_i(\tilde{x}_i, u_i) = \int_{kT}^{kT+T} e^{\breve{A}_i(\tilde{x}_i, u_i)(kT+T-\tau)}\breve{B}_i d\tau$ and $C = \breve{C}$.

A velocity-based multiple model network in the discrete time domain is formed by weighing several velocity-based local models:

$$W(k+1) = \sum_i \rho_i(\tilde{\psi}_0)(\boldsymbol{\Phi}_i(\tilde{x}_i, u_i)W(k) + \boldsymbol{\Gamma}_i(\tilde{x}_i, u_i)u(k))$$

$$Y(k) = CW(k) \tag{5.32}$$

5.4.3 Case Study

One of the key issues in this section is to demonstrate the effectiveness of the developed approach above by simulation. For simplicity, the CSTR (Continuous Stirred Tank Reactor) process is employed here.

5.4.3.1 Implementation of VB Multiple Model Networks

In simulation, we employ sinusoids and constant signals to approximate the step changes. This approximation simplifies the realization of the differential input signal for the VB multiple model networks in the continuous time domain. In the discrete time domain, the differential of the signal is deduced from

$$\frac{\Delta u(kT)}{T} = \frac{u(kT) - u(kT - T)}{T}$$

where T represents the sample time. This equation exactly matches the definition of the differential. There is no approximation at this stage. To transform the velocity-based LM network to a discrete-time model, the sampling time is selected as 0.1 min according to Shannon's sampling theorem.

5.4.3.2 Simulation Results

In this section, the simulation will be done in two parts. To get a clear idea about the performance of all the kinds of multiple models we discussed, we choose the same set of steps signal $q_c(t)$, which varies from 88 l/min to 110 l/min as shown in Figure 5-13. Firstly, continuous-time outputs from the velocity-based multiple model networks are compared with the corresponding discrete-time outputs; secondly, both of the outputs from the conventional LM network and the VB multiple models, in the discrete time domain, are compared with the output from the CSTR simulated process. Meanwhile, the modelling error from the velocity-based LM network and the conventional LM network are compared.

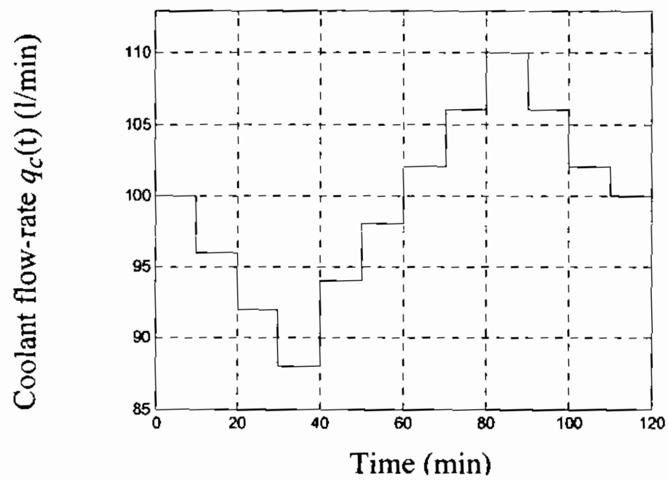
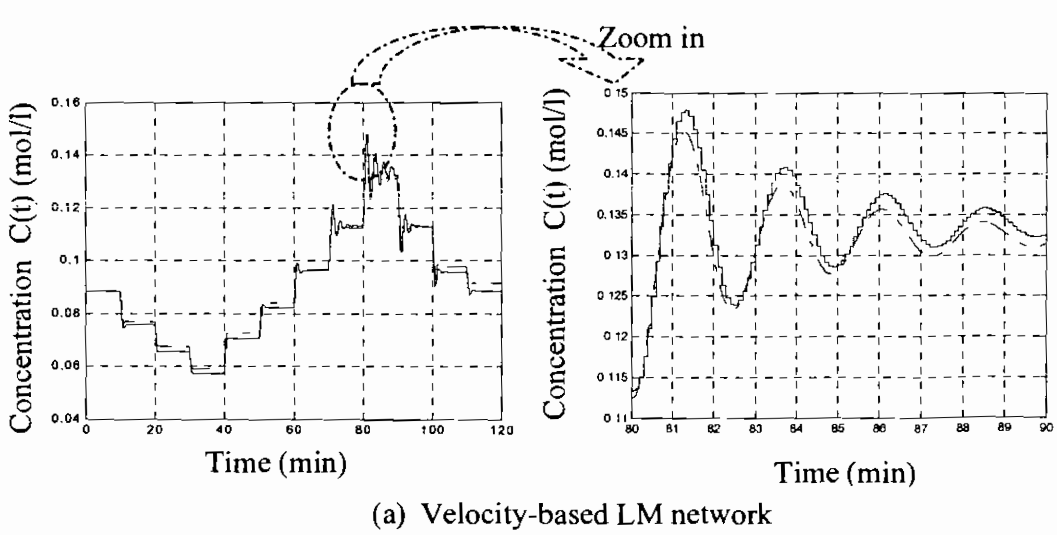


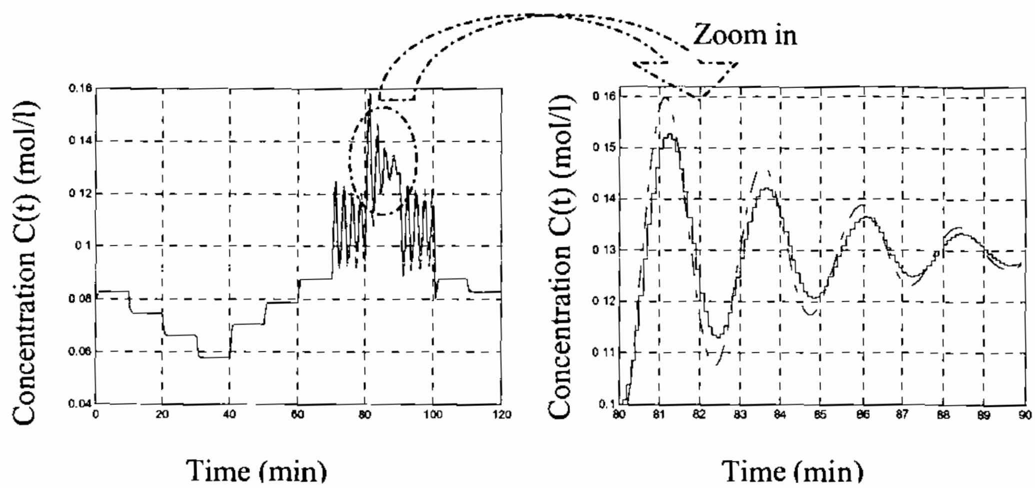
Figure 5-13 Step changes in coolant flow-rate $q_c(t)$

A. Comparison of the concentration outputs from the discrete-time models with the corresponding outputs from the continuous-time models.

Figure 5-14 shows that the dynamics of the discrete-time model matches those of the continuous-time model with certain accuracy. The results validate the use of the proposed continuous-to-discrete model transform approach. It should be noted that the steady-state error in the discrete-time velocity-based LM network is not equal to that of the continuous-time velocity-based LM network. This is because modelling errors exist in both the continuous-time and discrete-time velocity-based LM network. These errors accumulate with time. More detailed information is shown in Figure 5-16.



(a) Velocity-based LM network



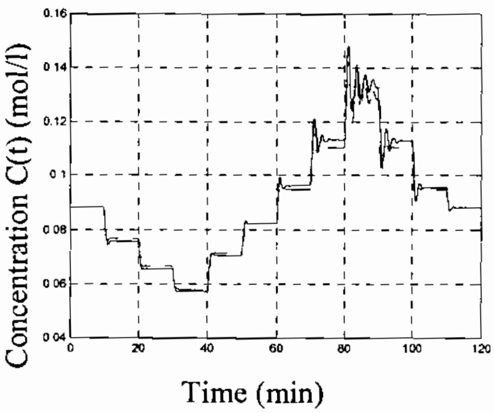
(b) Conventional LM network

Figure 5-14 Comparison of the concentration outputs

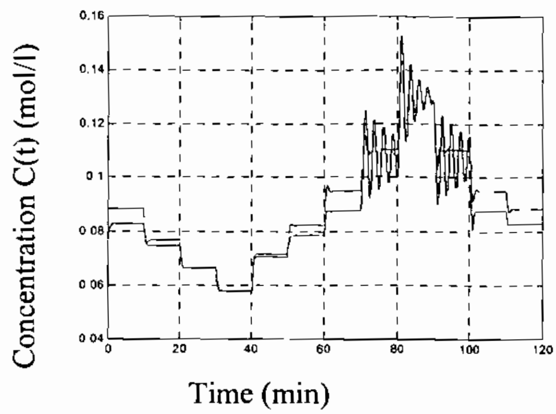
The solid line represents the outputs from the discrete time model, and the dash-dot line represents the outputs from the continuous time model.

B. Comparison of concentration outputs from the discrete-time model networks with corresponding outputs from the CSTR simulated process.

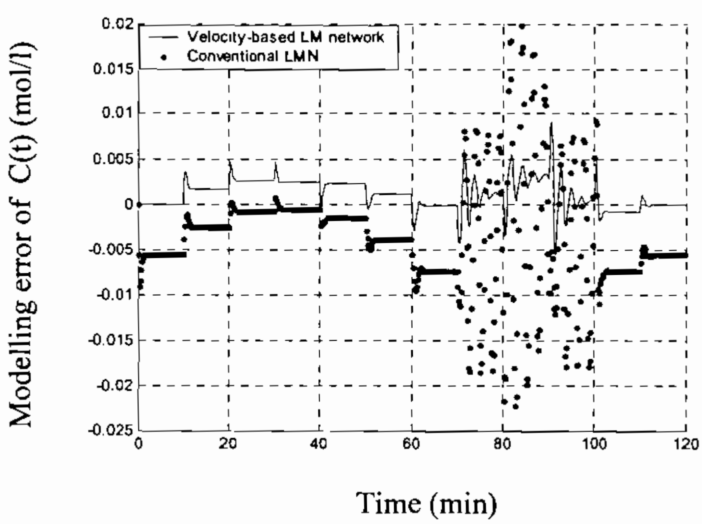
Figure 5-15 shows that the performances of both networks are relatively poor, especially in terms of steady-state accuracy. The discrete-time conventional LM network represents the CSTR simulated process accurately at points where only one model is valid (*C(t)*=0.058 *mol/l*). However, in the space between the models, the steady-state accuracy is relatively poor. Furthermore, the conventional LM network fails to capture the dynamics of the CSTR. The discrete-time velocity based LM network, on the other hand, shows much better capability in capturing the dynamics of the CSTR, especially when *C(t)* is about 0.11 mol/l - refer to the modelling error shown in Figure 5-15 (c).



(a) Velocity-based LM network



(b) Conventional LM network



(c) Modelling error

Figure 5-15 Comparison of concentration outputs.

In (a) and (b), the dashed line represents the output from the CSTR simulated process. The solid line represents the output from the discrete-time model networks. Modelling errors are shown in (c).

In summary, the proposed continuous-to-discrete time transform approach successfully represents the modelling property of continuous-time VB multiple model networks. The discrete-time concentration and temperature outputs from the velocity-based multiple network follow their corresponding continuous-time outputs and show better accuracy in modelling the system dynamics compared with those simulation outputs from the conventional LM network. However, the steady-state errors in the velocity-based multiple model networks are still significant, although the steady-state errors in the discrete-time domain are much smaller than those in the continuous-time domain.

Thus, there is a necessity to consider the steady-state error as an issue in the controller design. Section 5.3 proposed an 'adjustment' structure via the feedback of the plant output to the VB multiple model, which shows satisfactory trajectory tracking and regulation performance. Further work should consider the controller design based on discrete-time VB multiple model networks.

5.5 Concluding remarks

This chapter considered two issues based on a dynamic multiple model network structure (i.e. velocity-based multiple model networks) proposed recently (Leith and Leithead, 1998, 1999). The first issue dealt with the controller design based on the VB multiple model networks; the other issue concentrated on the Z-transform of VB multiple model networks.

The velocity-based analysis and design framework connects an instantaneous linearization with every operating point, which builds a direct relationship between the nonlinear system and the VB linearisation. Therefore, the velocity-based multiple model networks have an advantage in capturing the dynamics of nonlinear systems over the conventional LM networks. This provides a potential to design a better controller based on the dynamic information available from the VB multiple model networks. An integral state feedback controller is proposed, based on the VB multiple model networks.

The proposed controller design approach associates an integral action with state feedback in a local controller design, and then simply combines the local controllers together to give an overall system control signal. The method overcomes the difficulty in the implementation of the velocity-based approach, which normally involves deducing the differential signal. It exhibits continuity with linear control theory in the analysis of the overall system, which brings the potential to automatically design a controller with guaranteed performance and stability. Moreover, this approach shows satisfactory servo responses and much better regulator performance when compared with the proposed gain-scheduled LC network proposed in chapter 3. In addition, an approach for stability analysis was also proposed for the proposed controller design approach.

The second issue of this chapter is to develop the discrete velocity-based LM network. Both the velocity-based LM network and conventional LM network are transformed to the discrete time domain theoretically. Then, simulations on a highly nonlinear process (CSTR) show the effectiveness of the proposed continuous-to-discrete model transform approach and highlight that the velocity-based LM network has better capability in capturing the dynamics than the conventional LM networks. This brings promising potential for its application in controller design.

In summary, the velocity-based linearization approach proposed is new and promising for modelling and control of nonlinear systems. Its ability to accurately describe the complex nonlinear dynamics shows strong potential for the application of the model based controller design. There is a sound requirement for further research in this area, not only in the structure of the controller but also in the analysis and synthesis of the closed-loop systems. Meanwhile, some conventional controller design approaches for nonlinear systems, such as adaptive control and robust control, and model based predictive control have potential to be combined with VB multiple model networks. Further work should consider these issues.

Chapter 6

Conclusions and Future Work

6.1 Project Review

The project started by exploring linear model estimation techniques, such as ARX and ARMAX representations for systems with time delay. Algorithms that involve over-parameterisation of the process model, and those that involve approximating the time delay by a rational polynomial prior to identification were investigated. The results are reported in a paper (Gao and O'Dwyer, 2001), which discussed and compared two algorithms (for potential adaptive control development) theoretically and in simulation.

In practice, many real systems encountered exhibit complex nonlinearities that the conventional linear techniques couldn't handle. The last decade has shown a rapid increase in the use of local model representations of complex nonlinear dynamic systems. This approach employs multiple model networks for nonlinear system modelling and control; these networks consist of a family of subsystems and an interpolation scheme that orchestrates the interpolation between several subsystems.

The research has focused on the application of multiple model networks for control purposes. It concentrated on making a further step for automatically developing an overall global controller, with guaranteed stability and performance, based on well-developed linear controller design and analysis methods for complex dynamic nonlinear systems.

6.2 Concluding Remarks

The thesis has investigated and examined in detail the controller design approaches based on conventional local model networks and velocity-based multiple model networks. Both the local model network based GPC and LPV based GPC are employed to the control of nonlinear process CSTR in case study. An integral state feedback control based on velocity based multiple model networks is developed and successfully applied to the control of CSTR process. The main conclusions and contributions are summarised as follows:

Chapter 2:

- Theoretically, LM networks are capable of approximating nonlinear systems with certain accuracy by employing a finite number of linear (affine) local models, in practice, however, with limited number of local models, there are normally steady state errors involved in the network resulting from the normalization of validity functions.

- Case study on CSTR process shows employing affine LM networks is able to give satisfactory approximation accuracy, which has an extra freedom carried by the offset terms. However, the affine local model does not possess the superposition property fundamental to linear systems, which requires special consideration in controller design.

Chapter 3:

- Controller design based on LM networks is able to utilize the well-developed linear control theory to the control of nonlinear systems. It avoids the intensive computation load, such as the nonlinear optimisation to minimise the cost function for control, that the more usual approach based on a complete nonlinear model suffers.

- Assuming the system changes 'slowly' enough, introducing an integrative action with predictive control drives the closed-loop system output to the set point with good speed and accuracy for both the LC network and linear parameter variation (LPV) control.

- In the application of CSTR process control, both the LC network control based GPC and the linear parameter variation (LPV) control based GPC give globally satisfactory servo and regulator performance. There are no obvious advantages shown between them.

- Properly increasing the number of local model improves the global control performance, which benefits from better modelling accuracy. For the case of CSTR, both of the controller design approaches based on the LM network with two local models are able to give satisfactory servo and regulation performances.

- For the LC network based GPC, a satisfactory local controller performance doesn't naturally guarantee a satisfactory global controller performance, even though all the locally designed controllers satisfy the design criteria locally. This phenomenon is a problem with the method.

Chapter 4:

- Under the assumption that the nominal linear blending system is exponentially asymptotically stable, i.e. that there is a common positive definite matrix P existing for all the local systems, the corresponding affine blending system will be bounded by an ultimate value b, which is proportional to the maximum of the offset terms of the local systems.

- Introducing a state feedback controller with an extra term to control the affine blending systems. If the local systems have a common B, or if the weighting functions of the local models satisfy $\rho_i\rho_j = 0$, for $i \neq j$, $i, j = 1,\cdots,N$, the overall compensated system is stable at the origin; Generally, the overall compensated system is bounded by an ultimate bound b.

Chapter 5:

- The novel integral state feedback controller based on VB multiple model networks is simple to design and practical in application.
- The better and the more rigour the local controllers are designed, the better global performance can be expected, based on thc integral state feedback controller.
- The integral state feedback controller overcomes the difficulty in the implementation of the velocity-based approach, which normally involves deducing the differential signal.
- This integral state feedback controller holds the continuity with linear control theory in the analysis of the overall system, which brings the potential to automatically design a controller with guaranteed performance and stability.
- Moreover, the integral state feedback controller has an attractive advantage, which shows much better regulator performance when compared with the proposed gain-scheduled LC network proposed in chapter 3.
- A discrete-time version of the VB multiple model representation brings potential of VB multiple model networks application in the discrete time domain.

6.3 Future Work

The thesis has considered the control problem of hybrid dynamical systems in a particular case. We mean onc kind of interpolated systems i.e. LM/LC networks, consisting of a family of linear (or affine) subsystems and an interpolation scheme (typically formulated by employing Gaussian functions) that orchestrates the cooperation between them. In future work, whenever possible, problems should be formulated and discussed in the more general context of the switched system.

It remains true to declare that a deeper understanding of the behaviour of switched systems is crucial for obtaining an efficient solution to many real world control problems. In the context of the application in control of multiple model systems, some issues are of fundamental importance. Among these issues are:

- How to systematically design a controller, which guarantees global stability and desired servo and regulator performance?
- How weight functions in the interpolation scheme are reflected in controller and closed-loop system function modification?
- How to relax the stability condition for interpolated systems and also how to conquer the boundary problem?

In what follows, some proposals are made for the future work in these directions.

1. Systematic controller design and analysis with guaranteed stability and performance

Chapter 3 proposed three methods of controller design based on the interpolated multiple models. One of the common objectives in these methods, which require further development, is on a systematic controller design approach, which could automatically achieve stability and desired performance. The linear matrix inequality (LMI) based design technique has been effectively applied to achieve a global controller with guaranteed stability in recent years (Wang et al. 1996, Kim and Kim 2002, Rong, 2002).

Future research may be done in this area by extending these results to the case where performance is needed in addition to robustness and stability. One possibility is to use a guaranteed-cost framework developed, for example, in the model predictive control approach; an observer/controller system may be designed, which minimizes an upper bound on a quadratic performance index. Another extension of the existing work would be to include additive noise and develop a Kalman filter for the multiple model systems.

2. Stability analysis and control synthesis

On the stability analysis front, the approaches commonly used involve the application of the common Lyapunov function, multiple Lyapunov functions (Daafouz et al., 2002, Rubensson and Lennartson, 2000, Johansson et al., 1999, Tanaka et al., 2001) and interval approaches (Wang and Balakrishnan, 2002, Cao et al 1998).

In chapter 4, stability analysis on affine multiple models was given assuming the existence of a common Lyapunov function. However, it is not always easy to find a common positive definite matrix *P*, although there are some approaches available; for instance, the matrices 'commute' approach, the Lie algebra approach, another approach, for which that, for all $\lambda \in [0,1]$, $\lambda A_1 + (1-\lambda)A_2$ and $\lambda A_1 + (1-\lambda)A_2^{-1}$ are Hurwitz and finally, the LMI based approaches.

An alternative is to employ multiple Lyapunov functions. This approach designs one Lyapunov function for each controller and requires communication constraints. These constraints may be formulated as $V_i(x) \geq V_j(x)$ at points x where a switching from i to j can occur, i.e. $V_i(x)$ does not increase during transitions. This approach is more flexible compared with the common Lyapunov function approach. Issues to be investigated are:

- How this change affects the stability of the affine multiple systems?
- Is there a possibility to obtain a unified theory for stability analysis on switched systems?

On the synthesis front, efforts should concentrate on sound foundational and systematic formulations, as well as implementations of multiple models whose applications range from decision analysis to optimisation and control. Thus, issues pertaining to stability, controllability, observability, robustness, interpolation techniques (switching methods) should be addressed as they pertain to the behaviour of the synthesized model within a closed-loop control system.

3. Systematic controller design and analysis based on velocity-based multiple model networks

The novel velocity-based linearization exhibits strong potential in the representation of the relationship between the nonlinear systems and their instant velocity-based linearisation. One distinctive advantage of the velocity-based approach over conventional multiple models is that it is able to capture fast dynamic changes in nonlinear systems, which relaxes the requirement that the systems should change 'slowly' enough, which is required in conventional gain-scheduled systems. Further investigations should be performed to extend the proposed methods to more general nonlinear systems applications, for instance, applications in aviation, in which rapid dynamics are common.

Chapter 5 effectively applied the velocity-based multiple model networks to the controller design of the CSTR simulated process. Simulation results show strong robustness and improved performance. Although an approach was proposed on the stability analysis of the compensated system by using LMI techniques, more work is needed for the practical application of this method.

Meanwhile, some conventional controller design approaches for nonlinear systems, such as adaptive control and robust control, and model based predictive control have potential to be combined with VB multiple model networks. Not only the structure of the controller design but also the analysis and synthesis of the closed-loop systems need further investigation.

References

Agarwal, M. (1997). "A systematic classification of neural network based control", *IEEE Control Systems Magazine*, Vol.17, pp.75-93.

Åström K.J., Hagander, P. and Sternby, J. (1984). "Zeros of sampled systems", *Automatica*, Vol.20, No.1, pp.31-38.

Bergsten, P., Persson, M., Iliev, B. (2000). "Fuzzy gain scheduling for flight control", *Proceedings of 26^{th} Annual Conference of the IEEE Industrial Electronics Society (IECON 2000)*, Vol.1, pp. 271 –276.

Bhat, N.V. and McAvoy, T.J. (1990). "Use of neural nets for dynamical modelling and control of chemical process systems", *Computers and Chemical Engineering*, Vol.14, pp.573-583.

Billings, S.A. and Fakhouri, S.Y. (1982). "Identification of systems containing linear dynamic and static nonlinear elements", *Automatica*, Vol.18, pp.15-26.

Billings, S.A. and Voon, W.S.F. (1987). "Piecewise linear identification of non-linear systems", *International Journal of Control*, Vol.46, pp.215-235.

Boskovic, J. D. and Mehra, R. (1999). "Stable multiple model adaptive flight control for accommodation of a large class of control effector failures", in *Proceedings of American Control Conference*, San Diego, California, June, 1999, pp. 1920-1924.

Boyd, S., Hhaoui, L.E., Feron, E. and Balakrishnan V. (1994). Linear Matrix Inequality in Systems and Control Theory. Society for Industrial and Applied Mathematics, Philadelphia, USA.

Braun, M. W., Rivera, D.E. and Stenman, A. (2001). "A model-on-demand identification methodology for nonlinear process systems", *International Journal of Control*, Vol.74, No.18, pp.1708-1717.

Braun, M.W., D.E. Rivera, W. M. Carlyle, and Kempf, K.G. (2002). "A model predictive control framework for robust management of multi-Product, multi-echelon demand networks", *15th IFAC World Congress*, Barcelona, Spain, July 21-26.

Brown, M.D., Lightbody, G. and Irwin, G.W. (1997). "Nonlinear internal model using local model networks", *IEE Proceedings of Control Theory Applications*, Vol.144, No.6, pp. 505-514, November, 1997.

Brown M.D. and Irwin, G.W. (1999). "Nonlinear identification and control of turbogenerators using local model networks", *Proceedings of the American Control Conference*, San Diego, California, pp.4213-4217.

Cao, Y.Y., Lam, J. and Sun, Y.X. (1998). "Static output feedback stabilization: An ILMI approach", *Automatica*, Vol.34, No.12, pp.1641-1645.

Chan K.C., Leong, S.S. and Lin, G.C.I (1995). "A neural network PI controller tuner", *Artificial Intelligence in Engineering*, Vol.9, No.3, pp. 167-176.

Chen, S. and Billings, S.A. (1989). "Representation of nonlinear systems: the NARMAX model", *International Journal of Control*, Vol.49, No.3, pp.1013 -1032.

Chen, S., Billings, S.A. and Grant, P.M. (1990). "Nonlinear system identification using neural networks", *International Journal of Control*, Vol.51, No.6, pp.1191-1214.

Chen, C.L. and Chang, F.Y. (1996). "Design and analysis of neural/fuzzy variable structural PID control systems", *IEE Proceedings of Control Theory Application*, Vol.143, No.2, pp. 200-208.

Chen, S., Cowan, C.F.N. and Grant, P.M. (1991). "Orthogonal Least Squares Learning Algorithm for Radial Basis Function networks", *IEEE Transactions on Neural Networks*, Vol.2, No.2, pp.302-309.

Chow, C-M, Yoon, T.W. and Clarke, D.W. (1996). "An approach to multivariable predictive control with stability guarantees", Proceedings of United kingdom Automatic Control Conference (UKACC 1996), Oxford University Sep.2-5th, pp.1350-1355.

Clarke, D.W., Mohtadi, C. and Tuffs, P.S. (1987a). "Generalised Predictive Control-Part 1. The Basic Algorithm", *Automatica*, Vol.23, No.2, pp.137-148.

Clarke, D.W., Mohtadi, C. and Tuffs, P.S. (1987b). "Generalised Predictive Control-Part 2. The Basic Algorithm", *Automatica*, Vol.23, No.2, pp.149-160.

Clarke, D.W. and Mohtadi, C.(1989). "Properties of GPC", *Automatica*, Vol.25, pp.859-875.

Cutler, C.R. and Ramaker, B.L. (1980). "Dynamic matrix control-A computer control algorithm", *Proceedings of Joint American Control Conference (JACC)*, San Francisco, CS, pp.WP-5.

Cuzzola, F.A., Geromel, J.C. and Morari, M. (2002). "An improved approach for constrained robust model predictive control", *Automatica*, Vol.38, No.7, pp.1183-1189.

Cybenko, G. (1989). "Approximation by superposition of a sigmoidal function", *Mathematics of Control, Signals and Systems*, Vol.2, pp.303-314.

Daafouz, J., Riedinger, P. and lung, C. (2002). "Stability analysis and control synthesis for switched systems: A switched Lyapunov function approach", *IEEE Transactions on Automatic Control*, Vol.47, No.11, pp.1883-1887.

DeCarlo, R.A. (1989). Linear Systems, Prenticc Hall, Englewood Cliffs, N.J.

De Keyser, R.M.C. and Van Cauwenberghe, A.R. (1985). "Extended prediction self-adaptive control", *Proceedings of 7th IFAC Symposium on Identification and System Parameter Estimation*, York, U.K., pp.1255-1260.

Draeger, A., Engell, S., and Ranke, H. (1995). "Model predictive control using neural networks", *IEEE Control Systems Magazine*, Vol.15, No.5, pp.61-66.

Driankov, D., Palm, R. and Rehfuss, U. (1996). "A Takagi-Sugeno fuzzy gain-scheduler", *Proceedings of IEEE Conference on Fuzzy Systems*, New Orleans, pp.1053-1059.

Fantuzzi, C. and Rovatti, R. (1996). "On the approximation capabilities of the homogeneous Takagi-Sugeno model", *IEEE International Conference on Fuzzy Systems*, Vol. 2, pp.1067-1072.

Farinwata, S.S., Filev, D. and Langari, R. (2000), Fuzzy Control Synthesis and Analysis, John Wiley & Sons, LTD.

Filev, D.P. (1996). "Polytopic TSK fuzzy systems: analysis and synthesis", *Proceedings of the 5th IEEE International Conference on Fuzzy Systems*, pp.687-693.

Friedman, J.H. (1991). "Multivariable adaptive regression splines", The Annals of Statistics, vol.19, pp.1-141.

Fujinaka, T. and Omatu, S. (1999). "A switching scheme for adaptive control using multiple models", *IEEE SMC'99 Conference Proceedings of Systems, Man and Cybernetics*, Vol.5, pp.80-85.

Funahashi, K. (1989). "On the approximation realisation of continuous mappings by neural networks", Neural Networks, Vol.2, pp.183-192.

Gao R. and O'Dwyer A. (2001). "Comparison of two B-polynomial methods: identification of time delayed processes", *Proceedings of Irish Signals and Systems Conference (ISSC2001)*, Maynooth, Ireland, pp.383-388.

Gao R., O'Dwyer, A. and Coyle, E. (2002a). "A nonlinear PID controller for CSTR using local model networks", *Proceedings of the 4th World Congress on Intelligent Control and Automation*, June 10-14, Shanghai, P.R.China, pp.3278-3282.

Gao R., O'Dwyer, A. and Coyle, E. (2002b). "Model predictive control of CSTR based on local model networks", *Proceedings of Irish Signals and Systems Conference (ISSC2002)*, Cork, Ireland, pp.397-402.

Gawthrop, P.J., (1995). "Continuous-time local state local model networks", *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, Vancouver, Canada, pp.852-857.

Gawthrop, P.J., (1996). "Continuous-time local model networks", *Neural Adaptive Control Technology*, Singapore: World Scientific, pp.41-70.

Geng, G. and Geary, G.M. (1997). "Application of a neural network-based RLS algorithm in the generalised predictive control of a nonlinear air-handling plant", *IEEE Transaction on Control Systems Technology*, Vol.5, No.4, pp.439-445.

Gomi, H. and Kawato, M. (1993). "Neural networks for a closed-loop system using feed-back-error-control", *Neural Networks*, Vol.6, No.7, pp.933-946.

Grossberg, S. (1973). "Contour enhancement short term memory and constancy in reverberating neural networks", *Studies in Applied Mathematics*, Vol.52, pp.217-257.

Grossberg, S. (1978). A theory of visual coding, memory and development. Formal theories of visual perception, Wiley, New York.

Khalil, H.K. (1992), Nonlinear systems, Macmillan Publishing Company, New York.

Henson M.A and Seborg D.E. (1990), "Input-Output linearisation of general non-linear processes", *American Institute of Chemical Engineering (AIChE) Journal* Vol.36, pp.1753-1757.

Hilhorst, R. A., van Amerongen, J., Lohnberg, P. and Tulleken, H. J. A. F. (1991). "Intelligent adaptive control of mode-switch processes", *IFAC Symposia Series*, No.7, pp.145-150.

Hlavackova, K. (1996). "Dependence of the rate of approximation in a feedforward network on its activation function", *Neural Network World*, Vol. 6, No.2, p 163-171.

Hopfield, J.J (1982), "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of National Academy of Science of U.S.A.*, Vol.79, pp.2554-2558.

Hornik, K., Stinchcombe, M., and White, H. (1989), "Multilayer feedforward networks are universal approximators," *Neural Networks*, Vol.2, pp.359-366.

Hunt K.J., Sbarbaro D. (1991), "Neural networks for nonlinear internal model control", *Control Theory and Applications, IEE Proceedings D*, Vol.138, Issue 5, Sept.1991, pp.431-438.

Hunt K.J., Sbarbaro D., Zbikowski R. and Gawthrop P. J. (1992). "Neural networks for control systems- A survey", Automatica, Vol.28, No.6, pp. 1083-1112.

Hunt K.J., Haas, R., and Murray-Smith R. (1996). "Extending the functional equivalence of radial basis function networks and fuzzy inference systems", *IEEE Transactions on Neural Networks*, Vol.7, No.3, pp.776-781.

Hunt K.J. and Johansen T.A. (1997). "Design and Analysis of gain-scheduled control using local controller network", *International Journal of Control*, Vol.66, No.5, pp.619-651.

Irwin G.W., Warwick K., and Hunt K.J. (1995), "Neural network applications in control," The Institute of Electrical Engineering, London, United Kingdom.

Irwin, G.W., Brown, M.D., and Lightbody, G. (1998), "Nonlinear identification of a turbo-generator by local model networks," Special Session on Advances in Intelligent Control, *Proceedings of UKACC International Control Conference*, pp.1462-1467.

Jang J.S and Sun C.T. (1993). "Functional equivalence between radial basis function networks and fuzzy inference systems", *IEEE Transactions on Neural Networks*, Vol.4, No.1, pp.156-159.

Johansen, T.A and Foss B.A. (1993). "Constructing NARMAX models using ARMAX models", *International Journal of Control*, Vol.58, pp.1125-1153.

Johansen, T.A., and Foss, B.A. (1995). "Identification of nonlinear system structure and parameters using regime decomposition", *Automatica*, Vol31, pp.321-326.

Johansen T.A., Hunt K.J., P.J. Gawthrop and Fritz H. (1998 a). "Off-equilibrium linearization and design of gain scheduled control with application to vehicle speed control", *Control Engineering Practice*, Vol.6, No.2, pp.167-180.

Johansen T.A., Hunt K.J., and Fritz H. (1998 b). "A software environment for gain scheduled controller design", *IEEE Control Systems & Magazine*, April, pp.48-58.

Johansson M., Rantzer, A. and Arzen, K.E. (1999). "Piecewise quadratic stability of fuzzy system", *IEEE Transactions on Fuzzy systems*, Vol.7, pp.713-722.

Jury E.I. and Pavlidis T. (1963). "Stability and periodicity constraints for control system design", *IEEE Transactions on Circuits*, pp.128-141.

Kalvi, T. (1993). "ASMOD-an algorithm for adaptive spline modelling of observation data", *International Journal of Control*, Vol.58, No.4, pp.947-967.

Karimi. A. and Landau, I.D. (2000). "Robust adaptive control of a flexible transmission system using multiple models", *IEEE Transaction on Control Systems Technology*, Vol.8, No.2, pp.321-331.

Keerthi, S.S. and Gilbert, E.G. (1988), "Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving horizon approximations," Journal of Optimisation Theory and Applications, Vol. 57, No.2, pp.265-293.

Kim E. and Kim D. (2001), "Stability analysis and synthesis for an affine fuzzy systems via LMI and ILMI: Discrete case," *IEEE Transactions on Systems, Man, and Cybernetics,* Part B: Cybernetics, Vol31, No.1, pp.132-140.

Kim E. and Kim D. (2002), "Stability analysis and synthesis for an affine fuzzy systems via LMI and ILMI: Continuous case," *IEEE Transactions on Fuzzy System*, Vol.10, pp.391-400.

Kohonen, T (1982), "Self-organised formation of topologically correct feature maps," Biological Cybernetics, Vol.43, pp.59-69.

Korba, P.; Frank, P.M. (2000), "An applied optimization-based gain-scheduled fuzzy control," American Control Conference, 2000. Proceedings of the 2000, Vol.5, pp. 3383 -3387.

Kosko, B. (1994), "Fuzzy systems as universal approximators", IEEE Transactions on Computers, Vol. 43, n 11, pp.1329-1333.

Kouvaritakis, B. and Cannon, M. (2001), Nonlinear Predictive Control Theory and Practice, IEE Control Engineering Series 61, The Institution of Electrical Engineers, London, United Kingdom.

Leith D.J. and Leithead W.E. (1998), "Towards a theory of local model networks & blended multiple model systems," UKACC International Conference on Control, Sep.1-4, pp.509-514.

Leith D.J. and Leithead W.E. (1999) , "Analytic framework for blended multiple model systems using local linear models," International Journal of Control, Vol.72, pp.605-619.

Leith, D.J. Leithead, W.E. (2001), "Gain-scheduled control of a skid-to-turn missile: relaxing slow variation requirements by velocity-based design Proceedings of American Control Conference, pp.500-505.

Lightbody G. and Irwin G.W. (1997). "Nonlinear control structures based on embedded neural systems models", *IEEE Transaction on Neural Networks*, vol.8, No.3, pp.553-567.

Lin C.T. (1995). "A neural fuzzy control system with structure and parameter learning", *Fuzzy Sets and Systems*, Vol. 70, pp.183-212.

Lin Chun-Liang (2002). "On the design of an adaptive fuzzy gain-scheduled autopilot", Proceedings of the *American Control Conference*, 2002. Vol.2, pp.1567 -1572.

Li X., Wang, W. and Wang S. (2001). "Multi-model adaptive control for discrete-time systems", *Proceedings of the American Control Conference*, Arlington, VA June, pp.4820-4825.

Lu, Y. and Arkun Y. (2000), "Quasi-min-max MPC algorithms for LPV systems", *Automatica*, Vol.32, No.10, pp.527-540.

McConley, M.W., Appleby B. D., Dahleh, M.A. and Feron, E. (2002). "A computationally efficient Lyapunov-Based scheduling procedure for control of nonlinear systems with stability guarantees", IEEE Transaction on Automatic Control,Vol.45, pp.33-49.

McCulloch and Pitts (1943), "A logical calculus of the ideas Innmanent in Nervous Activity", *Bulletin of Mathematical Biophysics*, Vol.5, pp.115-137.

McLoone, J. and Irwin G.W. (1998a), "Continuous-time multiple model networks", *Proceedings of the Irish Signal and System Conference*, Dublin, Ireland, pp.67-74.

McLoone, J. and Irwin G.W., (1998b), "Process dynamic modelling using continuous-time local model networks", *Proceedings of the 5th IFAC Workshop on Algorithms and Architectures for Real-time Control*, Mexico, pp.179-184.

McLoone, S.C., "Nonlinear identification using local model networks". Ph.D. Thesis, The Queen's University of Belfast, 2000.

McLoone, S. C., Irwin, G. W. & McLoone, S. F. (2001). "Constructing networks of continuous-time velocity-based models," *IEE Proceedings of Control Theory and Applications*, 148 (5), September 2001, pp.397-405.

McLoone, S. C. & Irwin, G. W. (2001), "On nonlinear modelling using velocity-based multiple model networks", *Proceedings of the American Control Conference*, Arlington, June 25-27, Vol.2, pp.1622-1627.

Middleton, R.H.,Goodwin, G.C., Hill,D.J., and Mayne, D.Q., (1988), "Design issues in adaptive control", *IEEE Transaction on Automatic Control*, vol.33, no.1, pp.50-58.

Moody J. and Darken C. (1989). "Fast learning in networks of locally-tuned processing units", *Neural Computation*, Vol.1, pp.281-294.

Morari,M. and J.H.Lee (1999). "Model predictive control: past, present and future", *Computers and Chemical Engineering,* Vol.21, pp.965-980.

Morse, A.S., (1990). "Toward a unified theory of parameter adaptive control-tunability", *IEEE Transaction on Automatic Control*, vol.35, no.9, pp.1002-1012.

Morse, A.S., Mayne, D.Q., Goodwin, G.C. (1992), "Application of hysteresis switching in parameter adaptive control", *IEEE Transaction on Automatic Control*, vol.37, no.9, pp.1343-1354.

Murray-Smith, R., (1994). "A local model network approach to nonlinear modelling", Ph.D. Thesis, University of Strathclyde, U.K.

Murry-Smith R. and Johansen T.A. (1995). "Local learning in local model networks", *Proceedings of the Fourth International Conference on Artificial Neural Networks,* pp.40-46.

Murry-Smith R. and Johansen T.A. (1997), Local Learning in local model networks. Multiple Model Approaches to Modelling and Control. Chapter 7, Taylor and Francis.

Murry-Smith, R., Johansen T.A. and Shorten R. (1999). "On transient dynamics off-equilibrium identification in blended multiple model structures", European Control Conference, Karl Sruhe, Germany.

Narendra K.S. and Parthasarathy K. (1990), "Identification and control of dynamical systems using neural networks", IEEE transaction on neural networks, Vol.1, No.1, pp.4-27.

Narendra, K.S., and Balakrishnan, J. (1994 a) "Intelligent control using fixed and adaptive models", Decision and control, 1994, Proceedings of the 33rd IEEE conference on, vol.2, pp. 1680-1685.

Narendra, K.S., and Balakrishnan, J. (1994 b), "A common Lyapunov function for stable LTI systems with commuting A-Matrices", IEEE Transactions on Automatic Control, Vol.39, No.12, pp.2469-2471.

Narendra, K.S., Balakrishnan, J., and Ciliz, M.K., (1995). "Adaptation and learning using multiple models, switching, and tuning", *IEEE control systems magazine*, Vol.15, Issue:3, June, pp.30-42.

Narendra K.S. (1996), "Neural networks for control: theory and practice", *Proceedings of the IEEE*, Vol. 84, No.10, pp. 1385-1406.

Narendra K.S., and Balakrishnan, J. (1997). "Adaptive control using multiple models" *IEEE transaction on automatic control*, Vol.42, Issues:2, Feb., pp.171-187.

Narendra, K.S. and Cheng Xiang, (2000). "Adaptive control of discrete-time systems using multiple models", *IEEE transaction on Automatic control*, Vol.45, Sep., pp.1669-1686.

Nelles,O. (1997). "Orthonormal basis functions for nonlinear system identification with local linear model trees (LOLIMOT)", *Proceedings of 11th IFAC Symposium on system identification, (invited version)*, Japan, Vol.2, pp.667-672.

Nguyen, D.H. and Widrow B. (1990). "Neural networks for self-tuning control systems", *IEEE Control System Magazine*, Vol.10, pp.18-23.

Nichols R.A, Reichet R.T. and Rugh W.J., (1993). "Gain scheduling for H-infinity controllers: A flight control example," *IEEE Transaction Control Systems Technology*, Vol.1, No.2, pp.69-79.

Nϕ̈gaard M., Ravn O., Poulsen N.K., Hansen L.K. (2001), "Neural networks for modelling and control of dynamic systems," Springer-Verlag London.

Pischogios, D.C., De Veaux, R.D., and Ungar, L.H. (1992). "Nonparametric system identification: a comparison of MARS and neural nets", *Proceedings of the American control Conference*, Chicago, Illinois.

Psaltis, D., Sideris, A. and Yamamura, A.A. (1988). "A multiplayer neural network controller", *IEEE Control Magazines*, Vol.8, pp.17-21.

Psichogios, D.C. and Ungar, L.H. (1990). " Nonlinear internal model control and model predictive control using neural networks", *Proceedings of 5th IEEE International Symposium on Intelligent Control,* 5-9 Sept. pp.1082-1087.

Principe, J.C., Wang, L. and Motter, M.A. (1998), "Local dynamic modelling with self-organizing maps and applications to nonlinear system identification and control," IEEE Proceedings, Vol.86, No.11, pp.2240-2258.

Priestly, M.B. (1988). Non-linear and Non-stationary Time Series Analysis Academic Press.

Qin, S.J. and Badgwell, T.A. (1997), "An overview of industrial model predictive control technology", *Proceedings of the Fifth International Conference on Chemical Process Control*, AIChE, PP.232-256.

Richalet, J., Rault, A., Testud J.L. and Papon, J. (1978), "Model predictive heuristic control: applications to industrial processes," *Automatica*, Vol.14, No.5, pp.413-428.

Robison, B.D. and Clarke, D.W. (1991). "Robustness Effects of a Pre-filter in generalized predictive control," *IEE Proceedings*, Part D, Vol.138, pp.2-8.

Ronco, E., Arsan, T., Gawthrop, P.J. and Hill, D. J. (1999). "Globally valid continuous-time GPC through successive system linearizations", *Proceedings of the IEEE Conference on Decision and Control*, Vol.1, pp. 726-731.

Rong Q., Irwin G.W. and McLoone S. (2001). "Velocity-based local PID controller network", *Proceedings of Irish Signal and Systems Conference*, Maynooth, Ireland, pp.70-76.

Rong, Q. (2002), "Improved min-max MPC algorithm for LPV systems," *United Kingdom Automatic Control Conference (UKACC),* Postgraduate symposium, pp.167-173.

Rosenblatt, F. (1959), Principle of Neurodynamics, Spartan Books, New York.

Ruano A.E.B., Fleming P.J., Jones D.I (1992). "Connectionist approach to PID autotuning," *IEE Proceedings-D*, Vol.139, No.3, pp. 279-285.

Rubensson, M. and Lennartson, B. (2000). "Stability and robustness of hybrid systems using discrete time Lyapunov techniques", *Proceedings of America Control Conference*, Chicago, Illinois, pp.210-214.

Rugh W.J., (1991). "Analytical framework for gain scheduling", *IEEE Control Systems Magazine*, Vol.11, No.1, pp.79-84.

Rumulhart, D.E. and McClelland J.L. (1986), Parallel Distributed Processing, Vol.1, MIT press, Cambridge.

Sales, K.R. and Billings, S.A. (1990). "Self-tuning control of nonlinear ARMAX models", *International Journal of Control*, Vol.51, No.40, pp.753-769.

Schenker B. and Agarwal M. (1998). "Predictive control of a bench-scale chemical based on neural network models", *IEEE Transactions on Control Systems Technology*, Vol.6, No.3, pp.388-400.

Schetzen,M. (1981). "Nonlinear system modelling based on the wiener theory", *Proceedings of IEEE*, 69, (12), pp.1557-1573.

Scokaert, P.O.M. and Clarke, D.W. (1994). "Stabilizing Properties of Constrained predictive Control", *IEE Proceedings*, Part D, Vol.141, pp.295-304.

Seborg, D.E., Edgar, T.F. and Mellichamp, D.A. (1989), Process dynamics and control, Wiley Series in chemical Engineering.

Shamma, J.S. and Athans, M. (1990). "Analysis of gain scheduled control for non-linear plants", *IEEE Transactions on Automatic Control*, Vol.35., pp.898-907.

Shamma, J. S. and Athans, Michael (1991). "Gain scheduling: Potential hazards and possible remedies", *Proceedings of the American Control Conference*, Vol. 1, pp. 516-521.

Sharma K.S., McLoone S. and Irwin G. (2002). "Genetic algorithms for model and local controller network design", *Proceedings of the American Control Conference*, Anchorage, AK May 8-10th, pp.1693-1698.

Shibata T., Fukuda T. (1994). "Hierarchical intelligent control for robot motions", *IEEE Transactions on Neural Networks*", Vol.5, No.5, pp.823-832.

Skeppstedt, A., Ljung, L., and Millnert, M. (1992). "Construction of composite models from observed data", *International Journal of Control*, Vol.55, pp.141-152.

Shorten R. and Murry-Smith R. (1994), "On normalising radial basis function networks", *Proceedings of Irish Neural Networks Conference*, University College Dublin.

Shorten R., Murry-Smith R., Bjorgan R. and Golee H. (1999). "On the interpretation of local models in blended multiple model structures", *International Journal of Control*, Vol.72, pp.620-628.

Shorten R.N. and Narendra K.S. (2002). "Necessary and sufficient conditions for the existence of a common quadratic Lyapunov function for a finite number of stable second order linear time-invariant systems", *International Journal of Adaptive Control Signal Processing*, Vol.16, No.10, pp.709-728.

Shorten R.N. and Narendra K.S. (1999). "Necessary and sufficient conditions for the existence of a common quadratic Lyapunov function in two stable second order linear systems", *Proceedings of America Control Conference*, pp.1410-1414.

Shorten R., Narendra K.S. and Mason O. (2003). "A result on common quadratic Lyapunov functions", *IEEE Transactions on Automatic Control*, Vol.48, No.1, pp.110-113.

Soeterboek, R. (1992), Predictive Control –A Unified Approach, Prentice Hall International Series in Systems and Control Engineering. Prentice Hall International (UK) Limited.

Takagi, T. and Sugeno, M. (1985). "Fuzzy identification of systems and its applications for modelling and control", *IEEE Transactions on Systems, Man and Cyberbetics*, Vol.15, pp.116-132.

Tanaka, T. and Sugeno, M. (1992). "Stability analysis and design of fuzzy control systems", *Fuzzy Sets Systems*, Vol.45, pp.136-156.

Tanaka, K., Hori, T. and Wang H.O. (2001). "A dual design problem via multiple Lyapunov functions", *Proceedings of IEEE International Fuzzy Systems Conference*, pp.388-392.

Tong, H., and Lim, K.S., (1980). "Threshold auto-regression, limit cycles and cyclical data", *Journal of the Royal Statistical Society*, B, Vol.42, pp.245-292.

Townsend, S., Lightbody, G., Brown M.D. and Irwin G.W. (1998). "Nonlinear dynamic matrix control using local model networks", *Proceedings of American Control Conference*, Philadelphia, pp.801-805.

Townsend, S., and Irwin G.W., (1999). "Local model networks for nonlinear predictive control", *Proceedings of the 7th European Congress on Intelligent Techniques and Soft Computing*, Aachen, Germany, Sept. pp.13-16.

Vaccaro, R.J. (1995), Digital Control-A State-Space Approach, McGraw-Hill, Inc. International editions 1995.

Vachkov G. and Fukuda T. (1999). "Improved learning of fuzzy models by structured optimisation", *Proceedings of IEEE International Symposiums on Industrial Electronics*, Vol.3, pp.1135-1140.

VanDoren, V. (2001). "Model free adaptive control", *Control Engineering*, Europe, Feb/Mar, pp.25-31.

Wang, F., Balakrishnan, V. (2002). "Improved stability analysis and gain-scheduled controller synthesis for parameter-dependent systems", *IEEE Transactions on Automatic Control*, Vol.47, No.5, pp. 720 –734.

Wang, L. and Mendel, J.M. (1992). "Fuzzy basis functions, universal approximation, and orthogonal least-square learning", *IEEE Transactions on Neural Networks*, Vol.3, No.5, pp.807-814.

Wang, H.O., Tanaka, K., Michael, F. and Griffin, F. (1996). "An approach to fuzzy control of nonlinear systems: Stability and design issues", *IEEE Transaction on Fuzzy Systems*, Vol.4, No.1, pp.14-23.

Wang, L.-X., (1993). "Stable adaptive fuzzy control of non-linear systems", *IEEE Transactions on Fuzzy Systems*, Vol.1, pp.146-155.

Watanabe, R.; Uchida, K. (2002). "Adaptive identification as gain scheduling," *Proceedings of the American Control Conference*, Vol.2, pp.1532 –1537.

Weller, S.R, and Goodwin, G.C., (1994). "Hysteresis switching adaptive control of linear multivariable systems", *IEEE Transactions on Automatic Control*, Vol.39, no.7, pp.1360-1375.

Werbos, P. (1974). "Beyond regression new tools for prediction and analysis in the behavioural sciences," Ph.D. thesis, Harvard University, Cambridge, M.A.

Widrow, B. and Smith, F.W. (1964). "Pattern-recognizing control systems," *Proceedings of Computer and Information Science*, pp.46-53.

Widrow, B. and Stearns, S.D. (1985). Adaptive Signal Processing. Prentice Hall, Englewood Cliffs.

Widrow, B. and Lehr, M.A. (1990). "30 years of adaptive neural networks: Perceptron, Madaline, and Backpropagation," *Proceedings of The IEEE*, Vol.78, No.9, pp.1415-1442.

Wills M.J., Montague G.A., Massimo D.I. Tham, M.T. and Morris A.J. (1992). "Artificial neural networks in process estimation and control," *Automatica*, Vol.28, No.6, pp.1181-1187.

Ydstie, B.E. (1984). "Extended horizon adaptive control", *Proceedings of 9th IFAC World Congress*, Budapest, Hungary.

Yen J., Wang L. and Gillespie C.W. (1998). "Improving the interpretability of TSK fuzzy models by combining global learning and local learning", *IEEE Transactions on Fuzzy Systems*, Vol.6, No.4, pp.530-537.

Yu Wen-S and Lu T-C (1998). "PID controller designs using dynamical neural networks", *IEEE World Congress on Computational Intelligence, International Joint Conference on Neural Networks Proceedings*, Vol.3, pp. 2131-2135.

Zhang, J. and Morris, J. (2001). "Nonlinear model predictive control based on multiple local linear models", *Proceedings of the American Control Conference*, Arlington, VA June 25-27, pp.3503-3508.

Zhivoglyadov, P., Middleton, R.H. and Fu, M. (1997). "Localization based switching adaptive control for time varying discrete-time system", *Proceedings of 36th IEEE Conference on Decision and Control*, San Diego, pp.4151-4156.

Zeng, Xiao-Jun and Singh, M. G. (1994). "Approximation properties of fuzzy systems", *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol. 1, pp. 91-96.

Ziegler, J.G. and Nichols, N.B. (1942). "Optimum setting for automatic controllers", *Transaction on ASME*, Vo.64, pp.759-768.

Publications

Gao R. and O'Dwyer A. (2001), "Comparison of two B-polynomial methods: identification of time delayed processes", *Proceedings of Irish Signals and Systems Conference*, June 25th-27th, Maynooth, Ireland, pp.383-388.

Gao R., O'Dwyer A. and Coyle E. (2002), "A nonlinear PID controller for CSTR using local model networks", *Proceedings of the 4th World Congress on Intelligent Control and Automation*, June 10th-14th, Shanghai, P.R.China, pp.3278-3282.

Gao R., O'Dwyer A. and Coyle E. (2002), "Model predictive control of CSTR based on local model networks", *Proceedings of Irish Signals and Systems Conference*, June 24th-26th pp.397-402.

Gao R.,O'Dwyer A., McLoone S., and Coyle E. (2002), "Discrete-time velocity-based multiple model networks", *Proceedings of the 5th Portuguese Conference on Automatic Control*, pp.289-294.

Gao R., O'Dwyer A., McLoone S., and Coyle E. (2002), "Multiple model networks in non-linear system modelling for control-A review", *Wismarer Automatisierungs Symposium mit International Beteiligung, Hansestadt Wismar*, Sep. pp.13-3.

Gao R. (2002), "Velocity-based multiple model networks in discrete-time domain", Proceedings of UKACC, pp.72-77.

Gao R., O'Dwyer A., Coyle. E and McLoone S. (2003), "On the stability of affine blending systems", *Irish Signals and Systems Conference*, Cork, Ireland, pp.488-494.

Gao R, O'Dwyer A. (2004), "State feedback integral control by velocity-based multiple model networks", *Proceedings of American control conference*, Boston, Massachusetts, pp.2039-2044.