VELOCITY-BASED MULTIPLE MODEL NETWORKS IN THE DISCRETE TIME DOMAIN

Ruiyao Gao, Aidan O'Dwyer, Seamus McLoon*, Eugene Coyle

School of Control Systems and Electrical Engineering
Dublin Institute of Technology, Kevin Street,
Dublin 8, Ireland

*Department of Electronic Engineering,
National University of Ireland, Maynooth,
Co.Kildare, Ireland

Abstract: A conventional local model network (LMN) consists of a set of affined local models. It has poor interpretability on the process dynamic characters. Recent research on velocity-based multiple models show that the velocity-based approach is ideally suited to the development of local controller (LC) networks. As the applications of digital computer are popular in control, both the discrete-time conventional LM network and the discrete-time velocity-based multiple model approaches were developed theoretically. Simulation results on a complex nonlinear process continuous stirred tank reactor (CSTR) prove the effectiveness of the proposed approach.

Keywords: local model network, velocity-based multiple model network, nonlinear, discrete time domain.

1. INTRODUCTION

A static model gives information about the steady state relation between the input and the output signal. A dynamic model should give the relationships between the input and the output signal during transients. It is naturally much more difficult to capture dynamic behaviour. In an attempt to accurately model nonlinear dynamical systems, a wide variety of techniques have been developed such as nonlinear auto-regressive moving average with exogeneous inputs (NARMAX) models (Chen and Billings, 1989), Weiner models (Schetzen, 1981), Hammerstein models (Billings and Fakhouri, 1982) and Multiple Layer Perceptron (MLP) neural networks (Narendra and Kannan, 1990). However, all of these methods have difficulty in exploiting the significant theoretical results available in the conventional modelling because of their so-called black-box representation on dynamics of nonlinear systems. In contrast, Local Model Networks (LMN) can produce highly transparent empirical models. The locally valid sub-models are easily interpreted, and also the weighted sum of the local sub-models provides a qualitative high-level description of the nonlinear system.

However, recent research has questioned the ease of interpretability of the multiple model frameworks, demonstrating that dynamics of the LMN are only weakly related to the underlying local models. Leith and Leithead (1999) presented a novel class of blended multiple-model systems by which the global dynamics are directly related to the local models employed. The underlying sub-models are velocity-based, continuous and linear. The resulting continuity with existing linear techniques is useful for analysis and controller design. Further analytical results based on the complex nonlinear continuous stirred tank reactor (CSTR) process can be found in (McLoon, 2000). These results show that the velocity-based approach is ideally suited to the development of local controller networks.

Considering that the applications of digital computer are popular in the field of control, this paper aims to construct the computer oriented mathematical models for both the conventional LMN and Velocity-based LM Networks in discrete time domain. Simulation on highly nonlinear system CSTR will be studied to prove the effectiveness of the discrete models.

2. VELOCITY-BASED MULTIPLE MODEL NETWORKS IN CONTINUOUS TIME DOMAIN

Consider the general nonlinear state space system, with state vector x and input u :

$$\dot{x}(t) = f(x(t), u(t))$$
$$y(t) = g(x(t), u(t)) \tag{2.1}$$

For convenience, it is assumed that $y = Cx$ without loss of generality, because the output y is effectively a constant multiplied by the state vector. In many cases, the behaviour of a nonlinear system near an operating point (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:

$$x(t) = x_0 + \delta x(t)$$
$$u(t) = u_0 + \delta u(t) \tag{2.2}$$

where u_0 is nominal input and $\delta u(t)$ is the perturbation input. The input and state vector obey the differential equation, determined by submitting (2.2) into (2.1):

$$\delta \dot{x}(t) = f(x_0 + \delta x(t), u_0 + \delta u(t)) \tag{2.3}$$

Expanding the right-hand side of (2.3) in a Taylor series about (x_0, u_0) and keeping only the linear terms yields

$$\delta \dot{x}(t) = f(x_0, u_0) + \frac{\partial f}{\partial x}|_{(x_0,u_0)}\, \partial x(t) + \frac{\partial f}{\partial u}|_{(x_0,u_0)}\, \partial u(t) \tag{2.4}$$

Notice that $f(x_0,u_0)=0$. Defining $A=\frac{\partial f}{\partial x}|_{(x_0,u_0)}$, $B=\frac{\partial f}{\partial u}|_{(x_0,u_0)}$, we can rewrite eqn.(2.4) as

$$\delta\dot{x}(t)=A\partial x(t)+B\partial u(t) \tag{2.5}$$

Substituting Eqn.(2.2) into (2.4) and differentiating Eqn.(2.4) with respect to time gives the linear velocity-based system equation

$$\ddot{x}=\frac{\partial f}{\partial x}|_{(x_0,u_0)}\dot{x}+\frac{\partial f}{\partial u}|_{(x_0,u_0)}\dot{u} \tag{2.6}$$

With the appropriate initial conditions, eqn. (2.1) and eqn.(2.6) give identical solutions, and therefore there is no approximation at this stage. Eqn.(2.6) gives a direct relationship between the dynamics of velocity-based form of the nonlinear system and the velocity-based linearisation near an operating point. Furthermore, members of the family of velocity-based 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 linearised models as follows:

$$\ddot{x}=\left(\sum_i A_i(\tilde{x}_i,u_i)\rho_i(\tilde{\psi})\right)\dot{x}+\left(\sum_i B_i(\tilde{x}_i,u_i)\rho_i(\tilde{\psi})\right)\dot{u} \tag{2.7}$$

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 linearisation or freezing point of the *ith* local model:

$$\ddot{x}=A_i(\tilde{x}_i,u_i)\dot{x}+B_i(\tilde{x}_i,u_i)\dot{u} \tag{2.8}$$

The normalised weighting function is given by $\rho_i(\tilde{\psi})$, where $\tilde{\psi}$ is the scheduling vector.

The dynamics of the blended system, about the operating point $(\tilde{x}_0,u_0)$ is now considered. The velocity-based linearised form of Eqn.(2.7), at $(\tilde{x}_0,u_0)$, is simply obtained by freezing the validity function $\rho_i(\tilde{\psi})$ at the operating point to produce the following linear system:

$$\ddot{x}=\left(\sum_i A_i(\tilde{x}_i,u_i)\rho_i(\tilde{\psi}_0)\right)\dot{x}+\left(\sum_i B_i(\tilde{x}_i,u_i)\rho_i(\tilde{\psi}_0)\right)\dot{u} \tag{2.9}$$

With the appropriate initial conditions, the solution to eqn.(2.9) is initially tangential to the solution of the velocity-based multiple model system in eqn.(2.7). 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 eqn.(2.9) as

$$\ddot{x}=\sum_i \rho_i(\tilde{\psi}_0)\left(A_i(\tilde{x}_i,u_i)\dot{x}+B_i(\tilde{x}_i,u_i)\dot{u}\right) \tag{2.10}$$

which clearly highlights this direct relationship between the frozen-form eqn.(2.9) of the velocity-based blended system and the underlying local models eqn.(2.10) 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. No such direct relationship exists between the dynamics of the conventional multiple model representation and the dynamics of the first-order expansion system. Further detailed theoretical analysis of both conventional and velocity-based nonlinear representations can be found in [5,6].

3. VELOCITY-BASED MULTIPLE MODEL NETWORKS IN DISCRETE TIME DOMAIN

3.1. NORMAL LOCAL MODEL NETWORK DEVELOPMENT IN DISCRETTE TIME DOMAIN

3.1.1. ZOH EQUIVALENT MODEL DEVELOPMENT

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

$$u(t)=u[k],\ kT\le t<kT+T \tag{3.1}$$

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

$$y[k]=y[kT] \tag{3.2}$$

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:

$$\begin{aligned}\dot{x}(t)&=Ax(t)+bu(t)\\ y(t)&=cx(t)+du(t)\end{aligned} \tag{3.3}$$

Because (3.3) is a first order differential equation, if the value of $x(t)$ is known at some time t_0, then the value of $x(t)$ at future times is given by

$$x(t)=e^{A(t-t_0)}x(t_0)+\int_{t_0}^{t}e^{A(t-\tau)}bu(\tau)d\tau \tag{3.4}$$

where the symbol e^{At} stands for the matrix exponential function. If $t_0=kT$ and $t=kT+T$, where T is sampling time, then (3.4) gives an update formula for the state vector at sampling instants. That is, integrating the state equation over one sample period yields

$$\begin{aligned}x(kT+T)&=e^{A(kT+T-kT)}x(kT)\\&\quad+\int_{kT}^{kT+T}e^{A(kT+T-\tau)}bu(\tau)d\tau\end{aligned} \tag{3.5}$$

Now recall from (3.1) that in the interval of integration, the function $u(t)$ is equal to $u[k]$, a constant. This constant can be taken outside of the integral as follows:

$$x(kT+T)=e^{AT}x(kT)+\left[\int_{kT}^{kT+T}e^{A(kT+T-\tau)}bd\tau\right]u[k] \tag{3.6}$$

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

$x[k]=x[kT]$,

$\Phi = e^{AT}$,

$$\Gamma = \int_{kT}^{kT+T} e^{A(kT+T-\tau)} b d\tau \quad (3.7)$$

Then (3.6) becomes the discrete time state space sequence by

$$x[k+1] = \Phi x[k] + \Gamma u[k] \quad (3.8)$$

Note that Γ in (3.8) is a constant vector. Also, using the output equation of (3.3), we can write

$$y[k] = cx[k] + du[k] \quad (3.9)$$

Equation (3.8) and (3.9) 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 a linear time invariant, then its ZOH equivalent will also be linear and time invariant.

3.1.2. NORMAL LM NETWORK DEVELOPMENT

Assuming we have a set of linearized local models for a nonlinear system described as eqn.(2.1); that is, each of them is governed as eqn.(2.4) and eqn.(2.5) by the following equations:

$$\begin{aligned} \delta\dot{x}(t) &= A_i(x_i,u_i)\delta x(t) + b_i(x_i,u_i)\delta u(t) \\ y(t) &= cx(t) \end{aligned} \quad (3.10)$$

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

According to the section 3.1.1, we have the ZOH equivalent models for each linearized model, as follows:

$$\begin{aligned} \delta\bar{x}[k+1] &= \Phi_i(x_i,u_i)\delta\bar{x}[k] + \Gamma_i(x_i,u_i)\delta u[k] \\ y[k] &= c\bar{x}[k] \end{aligned} \quad (3.12)$$

in which $\delta\bar{x}(k) = \bar{x}(k) - x_{ie}$,

$\delta u(k) = u(k) - u_{ie}$, $\Phi_i = e^{A_i T}$,

$$\Gamma_i = \int_{kT}^{kT+T} e^{A_i(kT+T-\tau)} b_i d\tau$$

We can rewrite the Eqn.(3.12) as

$$\begin{aligned} \bar{x}[k+1] &= \Phi_i(x_i,u_i)(\bar{x}[k] - x_{ie}) \\ &\quad + \Gamma_i(x_i,u_i)(u[k] - u_{ie}) + x_{ie} \\ y[k] &= c\bar{x}[k] \end{aligned} \quad (3.13)$$

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

$$\begin{aligned} x[k+1] &= \sum_i \rho_i(\tilde{\psi}_0) \begin{pmatrix} \Phi_i(x_i,u_i)(x[k] - x_{ie}) \\ + \Gamma_i(x_i,u_i)(u[k] - u_{ie}) + x_{ie} \end{pmatrix} \\ y[k] &= cx[k] \end{aligned} \quad (3.14)$$

3.2. VELOCITY-BASED MULTIPLE MODEL NETWORKS DEVELOPEMENT

Recalling the linearized *ith* velocity-based local model in eqn.(2.8), we define $w = \dot{x}$. Rewriting it as follows:

$$\begin{pmatrix} \dot{\bar{x}} \\ \dot{\bar{w}} \end{pmatrix} = \begin{pmatrix} 0 & I \\ 0 & A_i(\tilde{x}_i,u_i) \end{pmatrix} \begin{pmatrix} \bar{x} \\ \bar{w} \end{pmatrix} + \begin{pmatrix} 0 \\ b_i(\tilde{x}_i,u_i) \end{pmatrix} \dot{u} \quad (3.15)$$

Then, the linearized model output is

$$\bar{y} = \begin{pmatrix} c & 0 \end{pmatrix} \begin{pmatrix} \bar{x} \\ \bar{w} \end{pmatrix} \quad (3.16)$$

For simplicity, we write the eqn (3.15) as follows

$$\begin{aligned} \dot{\breve{W}} &= \breve{A}_i(\tilde{x}_i,u_i)\breve{W} + \breve{B}_i(\tilde{x}_i,u_i)\dot{u} \\ \bar{y} &= \breve{C}\breve{W} \end{aligned} \quad (3.17)$$

in which, $\breve{A}_i(\tilde{x}_i,u_i) = \begin{pmatrix} 0 & I \\ 0 & A_i(\tilde{x}_i,u_i) \end{pmatrix}$,

$\breve{B}_i(\breve{x}_i,\breve{u}) = \begin{pmatrix} 0 \\ b_i(\tilde{x}_i,u_i) \end{pmatrix}$, $\breve{C} = \begin{pmatrix} c & 0 \end{pmatrix}$, $\breve{W} = \begin{pmatrix} \bar{x} \\ \bar{w} \end{pmatrix}$.

Then based on the section 3.1.1, we have the velocity based local state-space model

$$\begin{aligned} \breve{W}[k+1] &= \Phi_i(\tilde{x}_i,u_i)\breve{W}[k] + \Gamma_i(\tilde{x}_i,u_i)u[k] \\ \bar{y}[k] &= C\breve{W}[k] \end{aligned} \quad (3.18)$$

Where

$\Phi_i(\tilde{x}_i,u_i) = e^{\breve{A}_i(\tilde{x}_i,u_i)T}$,

$\Gamma_i(\tilde{x}_i,u_i) = \int_{kT}^{kT+T} e^{\breve{A}_i(\tilde{x}_i,u_i)(kT+T-\tau)} b_i d\tau$ and $C = \breve{C}$.

A velocity-based, blended, multiple model system in discrete time domain is formed by weighting several velocity-based local models:

$$\begin{aligned} W[k+1] &= \sum_i \rho_i(\tilde{\psi}_0)(\Phi_i(\tilde{x}_i,u_i)W[k] + \Gamma_i(\tilde{x}_i,u_i)u[k]) \\ Y[k] &= CW[k] \end{aligned} \quad (3.19)$$

4. CASE STUDY

4.1. CONTINUOUS STIRRED TANK REACTOR PROCESS

CSTR (Continuous Stirred Tank Reactor) is a highly non-linear process. A schematic of the CSTR system is shown in Figure 1. A single irreversible, exothermic reaction is assumed to occur in the reactor. The process model consists of two non-linear ordinary differential equations (McLoon,2000),

$$\begin{aligned} \dot{T}(t) &= \frac{q_f}{V}(T_f - T(t)) + K_1 C(t)\exp\left(-\frac{E}{RT(t)}\right) + \cdots \\ &\quad \cdots + K_2 q_c(t)\left[1 - \exp\left(-\frac{K_3}{q_c(t)}\right)\right](T_{cf} - T(t)) \\ \dot{C}(t) &= \frac{q_f}{V}(C_f - C(t)) - K_0 C(t)\exp\left(-\frac{E}{RT(t)}\right) \end{aligned}$$

$q_c(t)$ is the coolant flow rate, T(t) is the temperature of solution, $C(t)$ is the effluent concentration. The model parameters defined, and the nominal operating conditions are shown in table 1. The objective is to control $C(t)$ by manipulating $q_c(t)$.

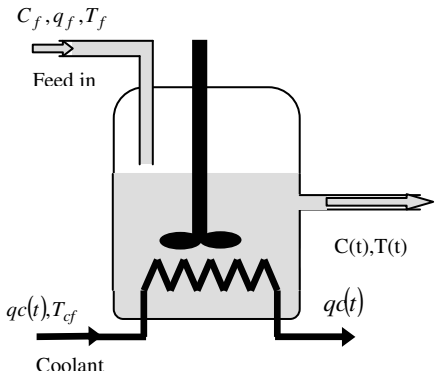


Fig. 1. Continuous Stirred Tank

Table 1. Nominal CSTR Operating Conditions

q_f = 100 l/min, product flow rate	C_f =1 mol/l,input concentration
T_f =350 K,input temprature	T_{cf} =350 K,temprature of coolant
K1=1.44*1013 Kl/min/mol,	V =100 l , container volume
E/R =104 K,activation energy	$K_2 = 0.01$ /l , constant
K3=700 l/min. constant	$K_0 = 7.2*10^{10}$ min^{-1} , constant

The CSTR plant is highly nonlinear with exponential terms and product terms. Furthermore, open-loop step tests show that the output concentration responses vary from over-damped to under-damped, indicating the variable dynamics in the CSTR process. Figure 2 is the loci of equilibrium distribution of input $q_c(t)$ versus output $C(t)$. The CSTR exhibits highly non-linear dynamical behaviour. Eigenvalue analysis shows that the stable equilibrium regime of the CSTR lies in $C(t) \in (0,0.13566) mol/l$ and $q_c(t) \in (0,111.0) l/\min$.

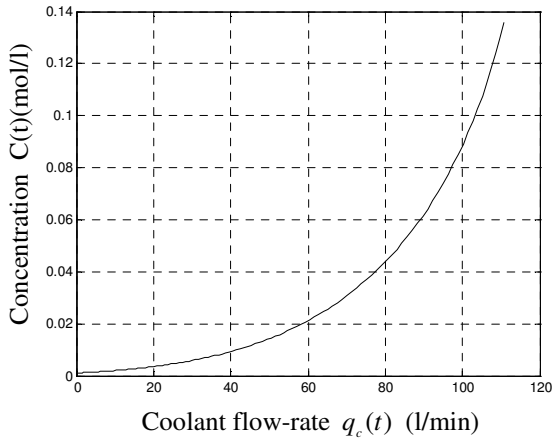


Fig 2 Loci of continuous stirred tank reaction

4.2. IMPLEMENTATION OF THE VELOCITY-BASED NETWORK

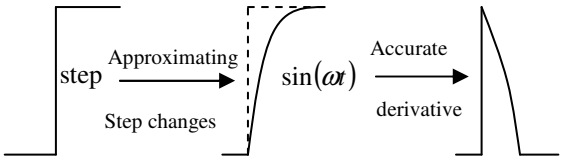
Recalling the eqn.(2.7), we see that the input of the velocity-based multiple model is the time differential of the control signal u. It is a pulse response model rather than a normal step response model. Practically, it is very difficult to formulate a pulse input signal because of the differential problem. Mathematically, in continuous time domain,

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

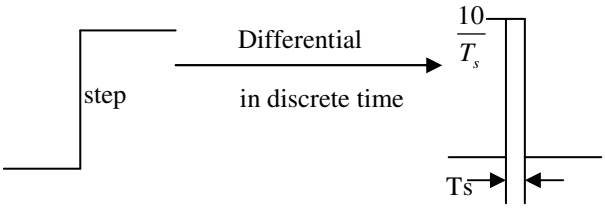
McLoon (2000) skilfully combined sinusoids and constant signals to approximate step changes, which can contribute a very good approximation of pulse signal for the velocity-based model. However, in the discrete time domain, the problem is simplified as

$$\dot{u}[kT] = \frac{u[kT] - u[kT-T]}{T},$$

which exactly matches the definition of the differential. There is no approximation at this stage. Figure 3 shows how the pulses are produced from step changes in both the continuous time domain and the discrete time domain.



(a) Using a sinusoid and step to formulate a pulse signal in the continuous time domain



(b) Pulse formulating in the discrete time domain

Fig.3 Pulse formulation

4.3. MODELLING THE CSTR PROCESS

The key issue of this paper is to deduce the conventional LM network and velocity-based LM network in discrete time domain and to show the effectiveness of the proposed approach by simulation. For simplicity, the authors employed the model structure applied successfully in (McLoon,2000). McLoon (2000) employed two local models to model the relationships between the coolant flow rate $q_c(t)$ and the product concentration, $C(t)$, for the operating space bounded by input: $q_c(t)$=[85, 111] l/min. These two local models are obtained by freezing the nonlinear velocity model at the appropriate linearisation 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 linearisation point of the *ith* local model.

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.

4.4. SIMULATION RESULTS

In this section, 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 step signal $q_c(t)$, which varies from 88 l/min to 110 l/min as shown in Figure 4. Firstly, continuous-time outputs from the velocity-based LM network are compared with the corresponding discrete-time outputs; meanwhile, continuous-time outputs from the conventional LMN are compared with their corresponding outputs in the discrete time domain. Secondly, both of the outputs from the conventional LMN and the velocity-based multiple models, in the discrete time domain, are compared with the output from the CSTR process.

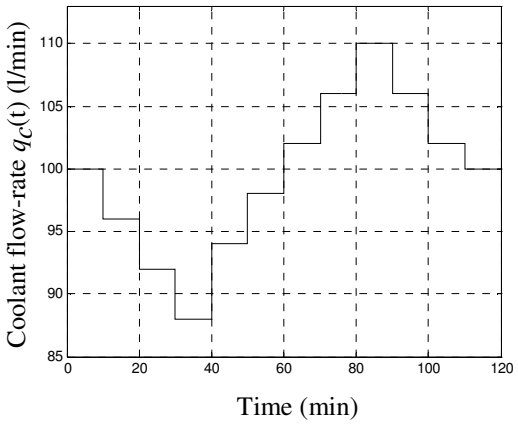
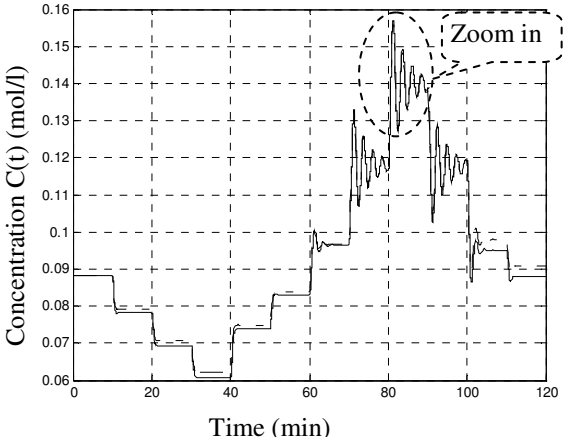
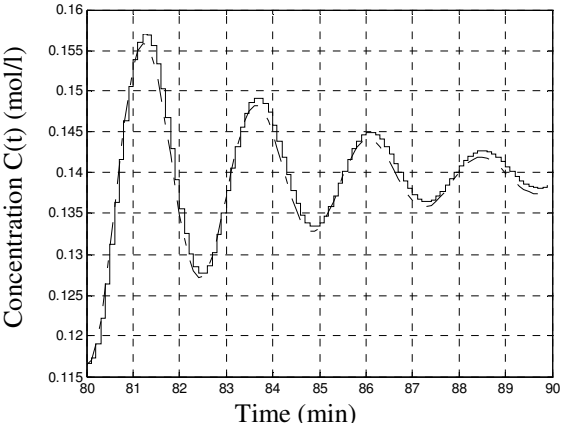


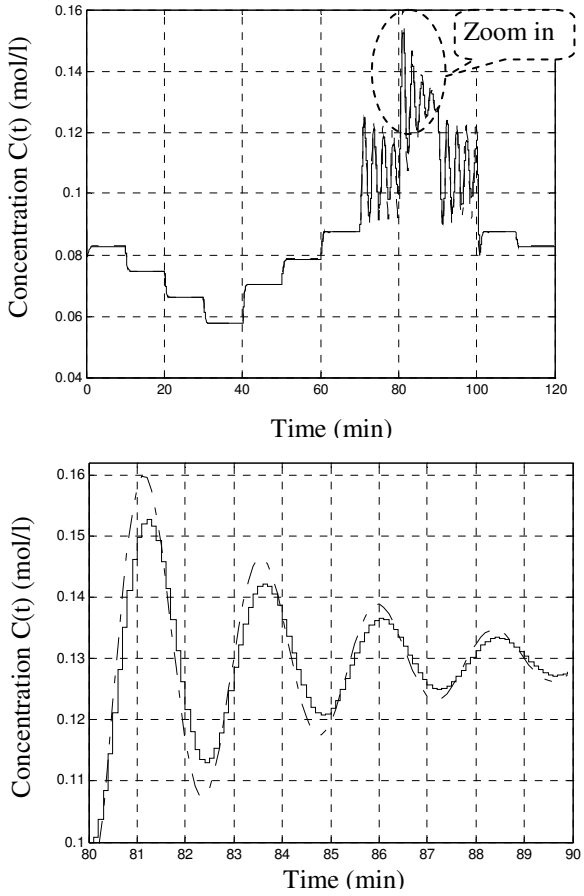
Fig.4 Step changes in coolant flow rate q_C(t)

A. Comparison of the concentration outputs from the models in the discrete time domain with those from the models in continuous time domain.





a) Velocity-based LM network



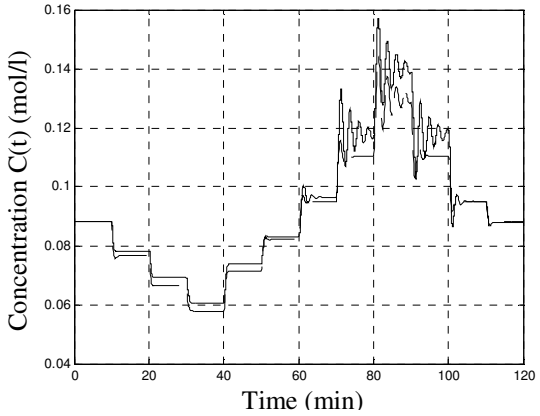
b) Conventional LM network

Fig.5. Comparison of the concentration outputs in continuous time domain with those in discrete time domain from Velocity-based LM network (a) and conventional LM network (b). Solid line represents the outputs from discrete time domain, and dash-dot line represents the outputs from the continuous time domain.

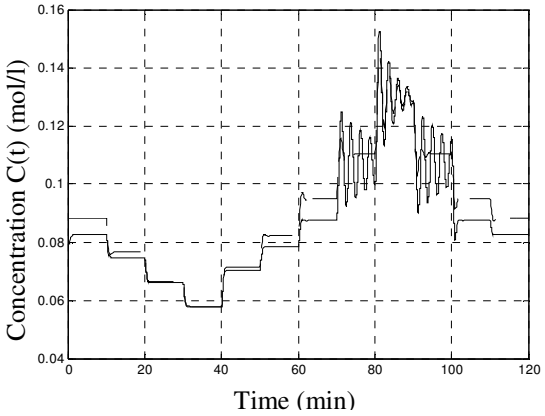
Fig.5. shows that the discrete model outputs follow the continuous models outputs and their matching is good in both types of model outputs. We zoom in on the highly dynamic areas to show the goodness of the match. The results prove the feasibility of the proposed continuous-to-discrete model transform

approach. One thing worthwhile to mention is that in Figure (a), the discrete-time velocity-based LM network doesn't exactly follow the continuous-time velocity-based LM network outputs, in the term of steady-state error. This is because modelling errors exist in both the continuous-time and discrete-time velocity-based LM network. These errors are out of the control of the network and therefore accumulate with time. More detailed information is shown in Figure 6.

B. Comparison of concentration outputs from the CSTR process with those from the velocity-based LM network and conventional LM network in discrete time domain.



a) Velocity-based LM network



b) Conventional LM network

Fig.6.Comparison of concentration output from the CSTR process with those from the velocity-based LM network in (a) and conventional LM network in (b) in discrete time domain. The dashed line represents the output from the CSTR process. The solid line represents the output from the models in discrete time domain.

Figure 6 shows that the performances of both networks are relatively poor, especially in terms of steady-state accuracy. The discrete-time conventional LM network does represent the CSTR plant accurately at points where only one model is valid. However, in the space between the models the steady-state accuracy is poor for the LM network is globally affined. This is very close to the analysis that Mcloon (2000) made on the models in continuous time domain. One thing should be noted is that the discrete-time velocity based LM network shows better capability in capturing the dynamics of CSTR process than the discrete-time conventional LM network, especially when C(t) is about 0.11 mol/l. However, steady state error is still significant, while the addition of more local models will improve the model accuracy as presented in (Gao, et al., 2002). Moreover, from the control point of view, the steady-state error can be compensated by introducing an integrator to the controller.

5. RESULTS

The main objective of this paper is to develop the discrete velocity-based LM network. In section 3, both of the velocity-based LM network and conventional LM network are transformed to the discrete time domain mathematically. Then simulations prove the effectiveness of proposed continuous-to-discrete model transform approach and highlight the better capability of the velocity-based LM network in capturing the dynamics of CSTR. Further work will focus on local controller networks design based on the developed discrete velocity-based LM network.

REFERENCES

Chen,S. and Billings, S.A., (1989), "Representation of nonlinear systems: the NARMAX model", International Journal of Control, 49, (3), pp.1013-1032.

Schetzen,M. (1981), "Nonlinear system modelling based on the wiener theory", Proceedings of IEEE, 69, (12), pp.1557-1573.

Billings,S.A. and Fakhouri, S.Y. (1982), "Identification of systems containing linear dynamic and static nonlinear elements", Automatica, 18, pp.15-26.

Narendra K.S. and Kannan P. (1990), "Identification and control of dynamical systems using neural networks", IEEE transaction on neural networks, Vol.1, No.1, pp.4-27.

Leith,D.J., and Leithead,W.E. (1999), "Analytic framework blended multiple model systems using linear local models", International Journal of Control, 1999,72,(7/8),pp.179-184.

McLoone,S.C. (2000), "Nonlinear identification using local model networks". Ph.D. Thesis, The Queen's University of Belfast, 2000.

Gao R., O'Dwyer, A. and Coyle,E. (2002), "A nonlinear PID controller for CSTR using local model networks", to appear in the 4th World Congress on Intelligent Control and Automation, in Shanghai, China, in June.