2002-01-01

Discrete-time velocity-based multiple model networks

Ruiyao Gao
Technological University Dublin

Seamus McLoone
NUI Maynooth

Eugene Coyle
Technological University Dublin

Proceedings of Controlo 2002: Portuguese Conference on Automatic Control, Aveiro, Portugal,
September 2002, pp. 289-294.

DISCRETE-TIME VELOCITY-BASED MULTIPLE MODEL NETWORKS

Ruiyao Gao, Aidan O'Dwyer, Séamus McLoone*, 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: The velocity-based local model (LM) network is a novel modelling approach that overcomes the lack of interpretability associated with the conventional LM network technique. The global dynamics of the nonlinear network are directly related to the underlying sub-model dynamics. Thus, the velocity-based network is ideally suited to the development of local controller (LC) networks. Furthermore, the local models are continuous-time, velocity-based and linear, providing continuity with established linear theory. To date, research has focused on the continuous-time version of the velocity-based network. The application of digital computer is widely popular in the field of control and, therefore, this paper develops a discrete-time velocity-based multiple model representation. A complex nonlinear process, in the form of a simulated continuous stirred tank reactor, is used to examine the modelling capabilities of the proposed discrete-time technique.

Keywords: multiple model networks, velocity-based networks, discrete-time nonlinear identification.

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 relationship 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 exogenous 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 of nonlinear systems.

In contrast, Local Model (LM) network was proposed as a modelling frame that could produce highly transparent models (Johansen, 1993). It was purported that the locally valid sub-models were easily interpreted and that the weighted sum of the local sub-models provided 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 the global dynamics of the conventional LM network are only weakly related to the dynamics of the underlying local models. Leith and Leithead (1999) presented a novel class of blended multiple-model networks whereby the global dynamics are directly related to the local models employed. Moreover, the underlying sub-models are continuous-time, velocity-based and linear, thus ensuring continuity with existing linear techniques, which is useful for analysis and controller design. Furthermore, analytical results based on the complex nonlinear continuous stirred tank reactor (CSTR) process show that the velocity-based approach is ideally suited to the development of local controller (LC) networks (McLoone, 2001).

So far, a lot work has been done regarding the conventional LM technique in both the continuous-time and discrete-time domains. However, all the studies relating to velocity-based LM networks exist in the continuous-time domain. Considering the popular applications of digital computer in the field of control and the potential capability of velocity-based LM network approach in the development of LC networks, this paper develops and presents a discrete-time version of the velocity-based multiple model representation. The modelling capabilities of the resulting nonlinear model are examined using a highly complex nonlinear process, in the form of a simulated continuous stirred tank reactor.

Section 2 briefly outlines the continuous-time velocity-based multiple network approach, while section 3 develops the novel discrete-time version of the network. Section 4 describes the CSTR process. The simulation results are given in section 5 while the paper ends with some conclusions and suggestions for future work in section 6.

2. CONTINUOUS-TIME VELOCITY-BASED MULTIPLE MODEL NETWORKS

Consider the general nonlinear state space system, with state vector $\mathbf{x}$ and input u :

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

For convenience, it is assumed that $\mathbf{y} = \mathbf{Cx}$ without loss of generality, because the output y is effectively a constant vector multiplied by the state vector. In many cases, the behaviour of a nonlinear system near an operating point $(\mathbf{x_0}, \mathbf{u_0})$ 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:

$$\mathbf{x}(\mathbf{t}) = \mathbf{x_0} + \mathbf{dx}(\mathbf{t})$$
$$\mathbf{u}(\mathbf{t}) = \mathbf{u_0} + \mathbf{du}(\mathbf{t}) \quad (2.2)$$

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

$$\mathbf{d\dot{x}}(\mathbf{t}) = \mathbf{f}(\mathbf{x_0} + \mathbf{dx}(\mathbf{t}), \mathbf{u_0} + \mathbf{du}(\mathbf{t})) \quad (2.3)$$

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

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

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

$$\mathbf{d\dot{x}}(\mathbf{t}) = \mathbf{A}\partial \mathbf{x}(\mathbf{t}) + \mathbf{B}\partial \mathbf{u}(\mathbf{t}) \quad (2.5)$$

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

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

With the appropriate initial conditions, (2.4) and (2.6) give identical solutions, and therefore there is no approximation at this stage. Equation (2.6) gives a direct relationship between the dynamics of velocity-based form of the nonlinear system and the velocity-based linearization near an operating point. Furthermore, members of the family of velocity-based linearization 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:

$$\ddot{\tilde{\mathbf{x}}} = \left(\sum_{\mathbf{i}} \mathbf{A_i}(\tilde{\mathbf{x}}_\mathbf{i}, \mathbf{u_i}) ?_\mathbf{i}(\tilde{?}) \right) \dot{\tilde{\mathbf{x}}} + \left(\sum_{\mathbf{i}} \mathbf{B_i}(\tilde{\mathbf{x}}_\mathbf{i}, \mathbf{u_i}) ?_\mathbf{i}(\tilde{?}) \right) \dot{\mathbf{u}} \quad (2.7)$$

where

$\mathbf{A_i}(\tilde{\mathbf{x}}_\mathbf{i}, \mathbf{u_i}) = \partial \mathbf{f} / \partial \tilde{\mathbf{x}} |_{(\tilde{\mathbf{x}}, \mathbf{u}) = (\tilde{\mathbf{x}}_\mathbf{i}, \mathbf{u_i})}$,

$\mathbf{B_i}(\tilde{\mathbf{x}}_\mathbf{i}, \mathbf{u_i}) = \partial \mathbf{f} / \partial \mathbf{u} |_{(\tilde{\mathbf{x}}, \mathbf{u}) = (\tilde{\mathbf{x}}_\mathbf{i}, \mathbf{u_i})}$ and $(\tilde{\mathbf{x}}_\mathbf{i}, \mathbf{u_i})$ is the linearization or freezing point of the *ith* local model:

$$\ddot{\tilde{\mathbf{x}}} = \mathbf{A_i}(\tilde{\mathbf{x}}_\mathbf{i}, \mathbf{u_i}) \dot{\tilde{\mathbf{x}}} + \mathbf{B_i}(\tilde{\mathbf{x}}_\mathbf{i}, \mathbf{u_i}) \dot{\mathbf{u}} \quad (2.8)$$

The normalised weighting function is given by $?_\mathbf{i}(\tilde{?})$, where $\tilde{?}$ is the scheduling vector.

The dynamics of the blended system, about the operating point $(\tilde{\mathbf{x}}_\mathbf{0}, \mathbf{u_0})$ is now considered. The velocity-based linearized form of (2.7), at $(\tilde{\mathbf{x}}_\mathbf{0}, \mathbf{u_0})$, is simply obtained by freezing the validity function $?_\mathbf{i}(\tilde{?})$ at the operating point to produce the following linear system:

$$\ddot{\tilde{\mathbf{x}}} = \left(\sum_{\mathbf{i}} \mathbf{A_i}(\tilde{\mathbf{x}}_\mathbf{i}, \mathbf{u_i}) ?_\mathbf{i}(\tilde{?}_\mathbf{0}) \right) \dot{\tilde{\mathbf{x}}} + \left(\sum_{\mathbf{i}} \mathbf{B_i}(\tilde{\mathbf{x}}_\mathbf{i}, \mathbf{u_i}) ?_\mathbf{i}(\tilde{?}_\mathbf{0}) \right) \dot{\mathbf{u}} \quad (2.9)$$

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

$$\ddot{\tilde{\mathbf{x}}} = \sum_{\mathbf{i}} ?_\mathbf{i}(\tilde{?}_\mathbf{0}) \left(\mathbf{A_i}(\tilde{\mathbf{x}}_\mathbf{i}, \mathbf{u_i}) \dot{\tilde{\mathbf{x}}} + \mathbf{B_i}(\tilde{\mathbf{x}}_\mathbf{i}, \mathbf{u_i}) \dot{\mathbf{u}} \right) \quad (2.10)$$

which clearly highlights this direct relationship between the frozen-form (2.9) of the velocity-based blended system and the underlying local models (2.10) at $(\tilde{\mathbf{x}}_\mathbf{0}, \mathbf{u_0})$. 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 (Leith and Leithead, 1999, McLoone,2001).

3. DISCRETE-TIME VELOCITY-BASED MULTIPLE MODEL NETWORKS

3.1. ZERO HOLD ORDER EQUIVALENT MODEL DEVELOPMENT

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

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

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

$$\mathbf{y}[\mathbf{k}] = \mathbf{y}[\mathbf{kT}] \quad (3.2)$$

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:

$$\dot{\mathbf{x}}(\mathbf{t}) = \mathbf{A}\mathbf{x}(\mathbf{t}) + \mathbf{B}\mathbf{u}(\mathbf{t})$$
$$\mathbf{y}(\mathbf{t}) = \mathbf{C}\mathbf{x}(\mathbf{t}) + \mathbf{D}\mathbf{u}(\mathbf{t}) \quad (3.3)$$

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

$$\mathbf{x}(\mathbf{t}) = \mathbf{e}^{\mathbf{A}(\mathbf{t}-\mathbf{t_0})}\mathbf{x}(\mathbf{t_0}) + \int_{t_0}^{t} \mathbf{e}^{\mathbf{A}(\mathbf{t}-\mathbf{t})}\mathbf{B}\mathbf{u}(\mathbf{t})\mathbf{dt} \quad (3.4)$$

where the symbol $\mathbf{e}^{\mathbf{At}}$ 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

$$\mathbf{x}(\mathbf{kT}+\mathbf{T}) = \mathbf{e}^{\mathbf{A}(\mathbf{kT}+\mathbf{T}-\mathbf{kT})}\mathbf{x}(\mathbf{kT}) + \int_{\mathbf{kT}}^{\mathbf{kT}+\mathbf{T}} \mathbf{e}^{\mathbf{A}(\mathbf{kT}+\mathbf{T}-\mathbf{t})}\mathbf{B}\mathbf{u}(\mathbf{t})\mathbf{dt} \quad (3.5)$$

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:

$$\mathbf{x}(\mathbf{kT}+\mathbf{T}) = \mathbf{e}^{\mathbf{AT}}\mathbf{x}(\mathbf{kT}) + \left[\int_{\mathbf{kT}}^{\mathbf{kT}+\mathbf{T}} \mathbf{e}^{\mathbf{A}(\mathbf{kT}+\mathbf{T}-\mathbf{t})}\mathbf{B}\mathbf{dt}\right]\mathbf{u}[\mathbf{k}] \quad (3.6)$$

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

$$\mathbf{x}[\mathbf{k}] = \mathbf{x}[\mathbf{kT}],$$
$$\mathbf{F} = \mathbf{e}^{\mathbf{AT}},$$
$$\mathbf{G} = \int_{\mathbf{kT}}^{\mathbf{kT}+\mathbf{T}} \mathbf{e}^{\mathbf{A}(\mathbf{kT}+\mathbf{T}-\mathbf{t})}\mathbf{B}\mathbf{dt} \quad (3.7)$$

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

$$\mathbf{x}[\mathbf{k}+\mathbf{1}] = \mathbf{F}\,\mathbf{x}[\mathbf{k}] + \mathbf{G}\mathbf{u}[\mathbf{k}] \quad (3.8)$$

Note that $\mathbf{G}$ in (3.8) is a constant vector. Also, using the output equation of (3.3), we can write

$$\mathbf{y}[\mathbf{k}] = \mathbf{C}\mathbf{x}[\mathbf{k}] + \mathbf{D}\mathbf{u}[\mathbf{k}] \quad (3.9)$$

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.

3.2. CONVENTIONAL *LM NETWORK DEVELOPMENT*

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

$$\mathbf{d\dot{x}}(\mathbf{t}) = \mathbf{A}_\mathbf{i}(\mathbf{x}_\mathbf{i},\mathbf{u}_\mathbf{i})\mathbf{dx}(\mathbf{t}) + \mathbf{B}_\mathbf{i}(\mathbf{x}_\mathbf{i},\mathbf{u}_\mathbf{i})\mathbf{du}(\mathbf{t})$$
$$\mathbf{y}(\mathbf{t}) = \mathbf{C}\mathbf{x}(\mathbf{t}) \quad (3.10)$$

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

According to the section 3.1, we have the ZOH equivalent models for each linearized model, as follows:

$$\mathbf{d\bar{x}}[\mathbf{k}+\mathbf{1}] = \mathbf{F}_\mathbf{i}(\mathbf{x}_\mathbf{i},\mathbf{u}_\mathbf{i})\mathbf{d\bar{x}}[\mathbf{k}] + \mathbf{G}_\mathbf{i}(\mathbf{x}_\mathbf{i},\mathbf{u}_\mathbf{i})\mathbf{du}[\mathbf{k}]$$
$$\mathbf{y}[\mathbf{k}] = \mathbf{C}\bar{\mathbf{x}}[\mathbf{k}] \quad (3.11)$$

in which $\mathbf{d\bar{x}}(\mathbf{k}) = \bar{\mathbf{x}}(\mathbf{k}) - \mathbf{x}_{\mathbf{ie}}$,

$\mathbf{du}(\mathbf{k}) = \mathbf{u}(\mathbf{k}) - \mathbf{u}_{\mathbf{ie}}$, $\mathbf{F}_\mathbf{i} = \mathbf{e}^{\mathbf{A}_\mathbf{i}\mathbf{T}}$,

$\mathbf{G}_\mathbf{i} = \int_{\mathbf{kT}}^{\mathbf{kT}+\mathbf{T}} \mathbf{e}^{\mathbf{A}_\mathbf{i}(\mathbf{kT}+\mathbf{T}-\mathbf{t})}\mathbf{B}_\mathbf{i}\mathbf{dt}$

We can rewrite the (3.11) as

$$\bar{\mathbf{x}}[\mathbf{k}+\mathbf{1}] = \mathbf{F}_\mathbf{i}(\mathbf{x}_\mathbf{i},\mathbf{u}_\mathbf{i})(\bar{\mathbf{x}}[\mathbf{k}] - \mathbf{x}_{\mathbf{ie}}) + \mathbf{G}_\mathbf{i}(\mathbf{x}_\mathbf{i},\mathbf{u}_\mathbf{i})(\mathbf{u}[\mathbf{k}] - \mathbf{u}_{\mathbf{ie}}) + \mathbf{x}_{\mathbf{ie}}$$
$$\mathbf{y}[\mathbf{k}] = \mathbf{C}\bar{\mathbf{x}}[\mathbf{k}] \quad (3.12)$$

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

$$\mathbf{x}[\mathbf{k}+\mathbf{1}] = \sum_\mathbf{i} \mathbf{?}_\mathbf{i}(\mathbf{?}_\mathbf{0}) \begin{pmatrix} \mathbf{F}_\mathbf{i}(\mathbf{x}_\mathbf{i},\mathbf{u}_\mathbf{i})(\mathbf{x}[\mathbf{k}] - \mathbf{x}_{\mathbf{ie}}) \\ + \mathbf{G}_\mathbf{i}(\mathbf{x}_\mathbf{i},\mathbf{u}_\mathbf{i})(\mathbf{u}[\mathbf{k}] - \mathbf{u}_{\mathbf{ie}}) + \mathbf{x}_{\mathbf{ie}} \end{pmatrix}$$
$$\mathbf{y}[\mathbf{k}] = \mathbf{C}\mathbf{x}[\mathbf{k}] \quad (3.13)$$

3.3. *VELOCITY-BASED MULTIPLE MODEL NETWORKS DEVELOPMENT*

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

$$\begin{pmatrix} \dot{\bar{\mathbf{x}}} \\ \dot{\bar{\mathbf{w}}} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{A}_\mathbf{i}(\tilde{\mathbf{x}}_\mathbf{i},\mathbf{u}_\mathbf{i}) \end{pmatrix} \begin{pmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{w}} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{B}_\mathbf{i}(\tilde{\mathbf{x}}_\mathbf{i},\mathbf{u}_\mathbf{i}) \end{pmatrix} \dot{\mathbf{u}} \quad (3.14)$$

Then, the linearized model output is

$$\bar{\mathbf{y}} = \begin{pmatrix} \mathbf{c} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{w}} \end{pmatrix} \quad (3.15)$$

For simplicity, we write the (3.14) as follows

$$\dot{\breve{\mathbf{W}}} = \breve{\mathbf{A}}_\mathbf{i}(\tilde{\mathbf{x}}_\mathbf{i},\mathbf{u}_\mathbf{i})\breve{\mathbf{W}} + \breve{\mathbf{B}}_\mathbf{i}(\tilde{\mathbf{x}}_\mathbf{i},\mathbf{u}_\mathbf{i})\dot{\mathbf{u}}$$
$$\bar{\mathbf{y}} = \breve{\mathbf{C}}\breve{\mathbf{W}} \quad (3.17)$$

in which, $\breve{\mathbf{A}}_\mathbf{i}(\tilde{\mathbf{x}}_\mathbf{i},\mathbf{u}_\mathbf{i}) = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{A}_\mathbf{i}(\tilde{\mathbf{x}}_\mathbf{i},\mathbf{u}_\mathbf{i}) \end{pmatrix}$,

$\breve{\mathbf{B}}_\mathbf{i}(\breve{\mathbf{x}}_\mathbf{i},\breve{\mathbf{u}}) = \begin{pmatrix} \mathbf{0} \\ \mathbf{b}_\mathbf{i}(\tilde{\mathbf{x}}_\mathbf{i},\mathbf{u}_\mathbf{i}) \end{pmatrix}$, $\breve{\mathbf{C}} = \begin{pmatrix} \mathbf{c} & \mathbf{0} \end{pmatrix}$, $\breve{\mathbf{W}} = \begin{pmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{w}} \end{pmatrix}$.

Then based on the section 3.1, we have the velocity based local state-space model

$$\breve{\mathbf{W}}[\mathbf{k}+\mathbf{1}] = \mathbf{F}_\mathbf{i}(\tilde{\mathbf{x}}_\mathbf{i},\mathbf{u}_\mathbf{i})\breve{\mathbf{W}}[\mathbf{k}] + \mathbf{G}_\mathbf{i}(\tilde{\mathbf{x}}_\mathbf{i},\mathbf{u}_\mathbf{i})\mathbf{u}[\mathbf{k}]$$
$$\bar{\mathbf{y}}[\mathbf{k}] = \mathbf{C}\breve{\mathbf{W}}[\mathbf{k}] \quad (3.18)$$

Where

$\mathbf{F}_\mathbf{i}(\tilde{\mathbf{x}}_\mathbf{i},\mathbf{u}_\mathbf{i}) = \mathbf{e}^{\breve{\mathbf{A}}_\mathbf{i}(\tilde{\mathbf{x}}_\mathbf{i},\mathbf{u}_\mathbf{i})\mathbf{T}}$,

$\mathbf{G}_\mathbf{i}(\tilde{\mathbf{x}}_\mathbf{i},\mathbf{u}_\mathbf{i}) = \int_{\mathbf{kT}}^{\mathbf{kT}+\mathbf{T}} \mathbf{e}^{\breve{\mathbf{A}}_\mathbf{i}(\tilde{\mathbf{x}}_\mathbf{i},\mathbf{u}_\mathbf{i})(\mathbf{kT}+\mathbf{T}-\mathbf{t})}\mathbf{b}_\mathbf{i}\mathbf{dt}$ and $\mathbf{C} = \breve{\mathbf{C}}$.

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

$$\mathbf{W}[\mathbf{k}+\mathbf{1}] = \sum_\mathbf{i} \mathbf{?}_\mathbf{i}(\mathbf{?}_\mathbf{0})(\mathbf{F}_\mathbf{i}(\tilde{\mathbf{x}}_\mathbf{i},\mathbf{u}_\mathbf{i})\mathbf{W}[\mathbf{k}] + \mathbf{G}_\mathbf{i}(\tilde{\mathbf{x}}_\mathbf{i},\mathbf{u}_\mathbf{i})\mathbf{u}[\mathbf{k}])$$
$$\mathbf{Y}[\mathbf{k}] = \mathbf{C}\mathbf{W}[\mathbf{k}] \quad (3.19)$$

4. CSTR 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 (Henson and Seborg, 1990),

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

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

$q_c(t)$ 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.

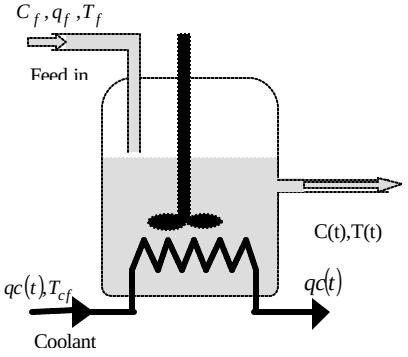


Fig. 1. Continuous Stirred Tank Reactor

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

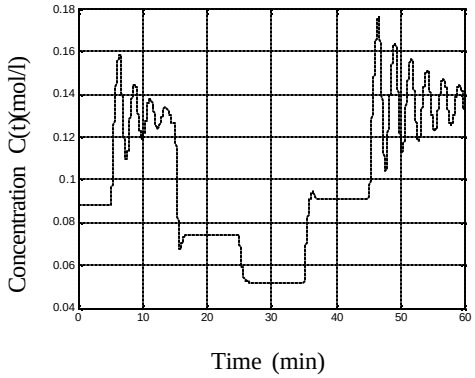


Fig 2 Dynamic response of the CSTR plant

The CSTR plant is highly nonlinear with exponential terms and product terms. Moreover, 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. Fig.2 is the step response of concentration output $C(t)$ when the coolant flow rate $q_c(t)$ varies from 85 l/min to 111 l/min. The CSTR exhibits highly non-linear dynamical behaviour.

5. MODELLING THE CSTR PLANT

5.1. IMPLEMENTATION OF THE VELOCITY-BASED NETWORK

Recalling the (2.10), 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,

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

and in the discrete time domain, the problem is simplified as

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

where T represents the sample time. This equation exactly matches the definition of the differential. There is no approximation at this stage. Figure 3 shows how the pulses are produced from step changes in the discrete time domain.

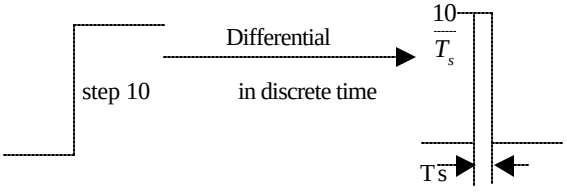


Fig.3. Discrete-time Pulse formulation

The key issue of this paper is to deduce the conventional LM network and velocity-based LM network in discrete time domain. For simplicity, the authors employed the strategy as (McLoone,2001), in which two local models are applied 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 linearization points:

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

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

in which (C_o^i, T_o^i, q_{co}^i) denotes the linearization point of the *ith* local model. Normalised Gaussian function

is commonly used as the weighting function to blend local models in both conventional LM network and velocity-based LM network. However, recent study (McLoone, 2001) pointed out its limited mapping accuracy in the special case CSTR and suggest to use piecewise linear weighting functions instead.

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.

5.2. 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 Fig.4. Firstly, continuous-time outputs from the velocity-based LM network are compared with the corresponding discrete-time outputs; Secondly, both of the outputs from the conventional LM Network and the velocity-based multiple models, in the discrete time domain, are compared with the output from the CSTR model plant. Meanwhile, the modelling error from the velocity-based LM network and the conventional LM network are compared too.

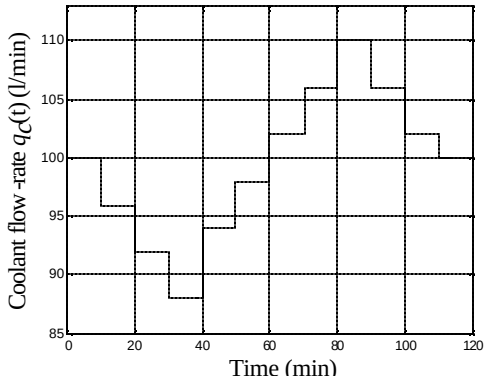
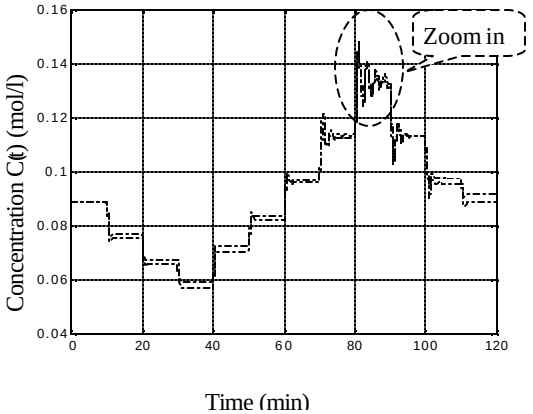


Fig.4. Step changes in coolant flow rate q_C(t)

A. Comparison of concentration outputs from velocity-based LM network in the discrete time domain with corresponding outputs in the continuous time domain.



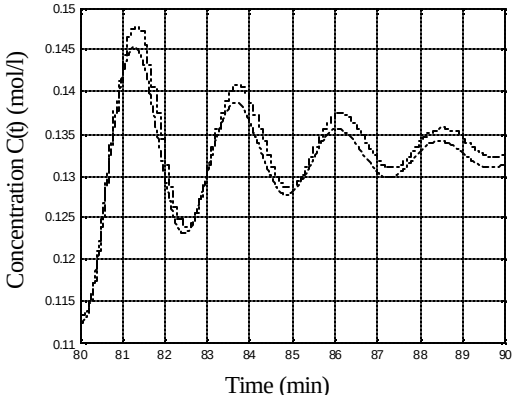
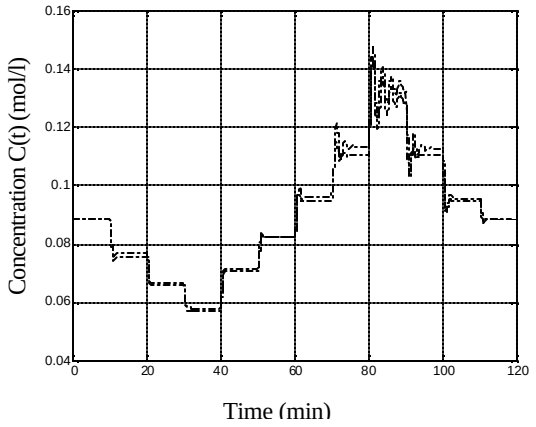


Fig.5. Comparison of the concentration outputs. Solid line represents the outputs from discrete-time model, and dash-dot line represents the outputs from the continuous-time model.

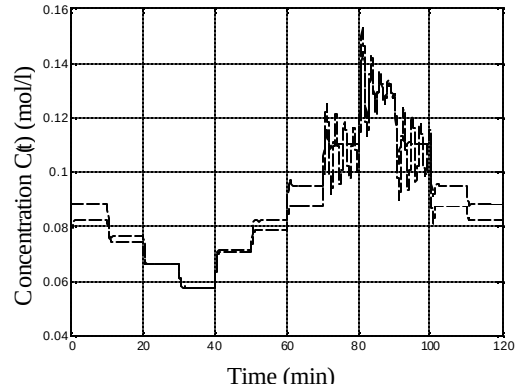
Fig.5. shows that the dynamics of the discrete-time model accurately matches those of the continuous-time model. The results validate the use of the proposed continuous-to-discrete model transform approach. It should be noted that the discrete-time velocity-based LM network doesn't exactly follow the continuous-time velocity-based LM network output in terms of the steady state. 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 Fig 6.

B. Comparison of concentration outputs from the discrete-time model networks with corresponding outputs from the CSTR plant model.

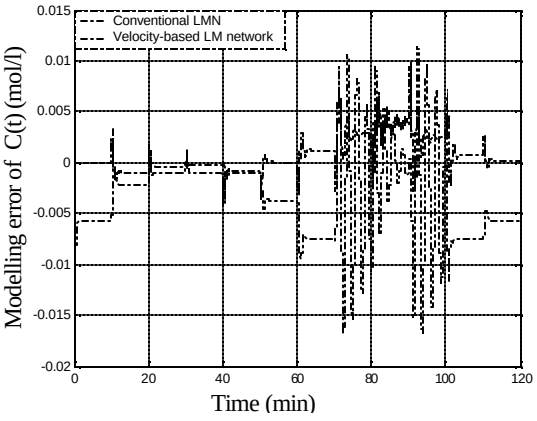
Fig. 6 shows that the performances of both networks are relatively poor, especially in terms of steady-state accuracy. The discrete-time conventional LM network represents the CSTR plant accurately at points where only one model is valid. However, in the space between the models the steady-state accuracy is poor for the LM network is globally affined. 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 CSTR plant, especially when C(t) is about 0.11 mol/l - refer to the modelling error shown in Fig. (c). Unfortunately, the steady state error is still significant. However, as discussed in (McLoone, 2001), this error is inherent in the velocity-based approach but can be removed using integral feedback in a control framework. Thus, the discrete-time velocity-based approach outlined in this paper is ideally suited to the development of discrete-time network consisting of a set of local controllers.



a) Velocity-based LM network



b) Conventional LM network



c) Modelling error

Fig.6. Comparison of concentration outputs. The dashed line represents the output from the CSTR plant. The solid line represents the output from the discrete-time models networks. The dash-dotted line represents the output from the CSTR plant model. Modelling errors are shown in (c).

6. CONCLUSIONS

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 on highly nonlinear plant CSTR prove 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. 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.

Henson M.A and Seborg D.E., (1990) 'Input-Output linearisation of general non-linear processes', AIChE Journal Vol.36, 1753-1757.

Johansen, T.A and Foss B.A., "Constructing NARMAX models using ARMAX models", International Journal of Control, Vol.58, 1993, pp.1125-1153.

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., Irwin,G.W. (2001), "On nonlinear modelling using velocity-based multiple model networks", 2001 American Control Conference (ACC'01), Arlington, June, pp.25-27.