IFAC Conference on New Technologies for Computer Control
19-22 November 2001, Hong Kong

BLENDING LOCAL MODELS WITHIN A VELOCITY-BASED MULTIPLE MODEL NETWORK

Séamus C. McLoone, George W. Irwin & Seán F. McLoone

Intelligent Systems and Control,
School of Electrical and Electronic Engineering,
Queen's University Belfast, Belfast, BT9 5AH, N. Ireland.
e-mail: seamus.mcloone@eeng.may.ie; g.irwin@ee.qub.ac.uk, s.mcloone@ee.qub.ac.uk

Abstract: Velocity-based local model (LM) networks overcome many of the disadvantages of the conventional LM network approach. In the former, a set of *velocity-based linear* local models are interpolated using appropriate weighting functions whereas the latter employs *affine* local models. In contrast to the conventional network, the global dynamics of the velocity-based nonlinear model comprise a simple linear combination of the local model dynamics. This paper examines in detail the blending of local models which is a key issue in the velocity-based approach. Various methods of blending the sub-models are studied and it is highlighted that only one of these is suitable. *Copyright © 2001 IFAC*

Keywords: Nonlinear, identification, multiple models, networks, interpolation.

1. INTRODUCTION

The goal of local model (LM) networks is to accurately represent a nonlinear dynamical plant over the entire operating space using a small number of local models. Since the greater part of the operating space generally consists of regions where no single model dominates, and several local models require blending, it is no surprise to discover that blending has a important role in multiple-model systems. For subsequent analysis and design it is desirable that the global dynamics of the nonlinear blended network are directly related to the underlying sub-model dynamics. Recent research (Shorten et al., 1999, Murray-Smith et al., 1999, Leith and Leithead, 1999) has highlighted the lack of this property within the conventional multiple model framework, demonstrating that, at best, the dynamics of the LM network are only weakly related to the underlying local models. Furthermore, existing multiple model systems typically employ affine models (McLoone and Irwin, 1998) which, being nonlinear, do not provide continuity with established linear theory and can also detract from the transparency of the overall network.

A novel class of velocity-based, blended multiple model system has been proposed by Leith and Leithead (1999) whereby the global dynamics are directly related to the local models employed. The underlying sub-models are velocity-based, continuous-time and linear, thus providing continuity with existing linear techniques, which is useful for analysis and controller design. Most of their work to date has concentrated on theoretical developments using simple illustrative examples.

This paper examines the key aspect of blending local models within the velocity-based multiple-model framework. A highly nonlinear dynamical system in the form of a simulated Continuous Stirred Tank Reactor (CSTR) process is used to analyse the

various methods of blending the velocity-based sub-models. Results reveal that there is only one correct blending technique for velocity-based networks.

The initial concept of LM networks (Johansen and Foss, 1992, 1993, Brown et al., 1997) envisaged weighting the outputs of each local model to obtain the global network output. In the case of a heterogeneous LM network (Murray-Smith, 1994) where local models can vary from complex neural networks to simple constants within one network, weighting the local model outputs is the most obvious, and possibly only, choice for interpolation. However, this paper highlights that blending in this manner is fundamentally incorrect for the weighting of local models within a velocity-based multiple-model system. Instead, the global model dynamics should be obtained by blending the local model dynamics.

For completeness the next section outlines the salient features of the velocity-based local modelling approach. In section 3 the different ways in which the velocity-based local models can be blended are described. Section 4 briefly describes the CSTR process, used for analytical purposes in this paper. Section 5 analyses the results of the various blending methods. The paper ends with some suggestions for future work in section 6.

2. VELOCITY-BASED LOCAL MODELLING

For ease of reference the notation of Leith and Leithead (1999) is generally followed in this section. Consider the general nonlinear state space system:

$$\dot{x} = f(x,u) \tag{1}$$

The first-order expansion of this system about an operating point (x_o,u_o) is given by:

$$\dot{\tilde{x}} = f(x_o,u_o) + \left.\frac{\partial f}{\partial x}\right|_{x_o,u_o}(\tilde{x}-x_o) + \left.\frac{\partial f}{\partial u}\right|_{x_o,u_o}(u-u_o) \tag{2}$$

Differentiating this *affine* expansion with respect to time gives the velocity-based *linear* system:

$$\ddot{\tilde{x}} = \left.\frac{\partial f}{\partial x}\right|_{x_o,u_o}\dot{\tilde{x}} + \left.\frac{\partial f}{\partial u}\right|_{x_o,u_o}\dot{u} \tag{3}$$

A similar form can be obtained by differentiating (1) with respect to time using the chain rule:

$$\ddot{x} = \frac{\partial f}{\partial x}\dot{x} + \frac{\partial f}{\partial u}\dot{u} \tag{4}$$

With the appropriate initial conditions, (1) and (4) give identical solutions and therefore there is no approximation at this stage. More importantly, however, is the fact that the velocity-based linearisation (3) is just the *frozen* form of the velocity-based system (4), at the operating point (x_o,u_o). This establishes a direct relationship between the dynamics of the velocity-based form of the nonlinear system and the velocity-based linearisation. Furthermore, members of the family of velocity-based linearisations are all linear, providing continuity with established linear theory and methods.

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

$$\begin{aligned}\ddot{\tilde{x}} &= \left(\sum_i A_i(\tilde{x}_i,u_i)\rho_i(\tilde{\psi})\right)\dot{\tilde{x}} \\ &+ \left(\sum_i B_i(\tilde{x}_i,u_i)\rho_i(\tilde{\psi})\right)\dot{u}\end{aligned} \tag{5}$$

where $A_i(\tilde{x}_i,u_i) = \left.\frac{\partial f}{\partial \tilde{x}}\right|_{\tilde{x}_i,u_i}$, $B_i(\tilde{x}_i,u_i) = \left.\frac{\partial f}{\partial u}\right|_{\tilde{x}_i,u_i}$ and $(\tilde{x}_i,u_i)$ is the linearisation or *freezing* point of the i^{th} local model:

$$\ddot{\tilde{x}} = A_i(\tilde{x}_i,u_i)\dot{\tilde{x}} + B_i(\tilde{x}_i,u_i)\dot{u} \tag{6}$$

The normalised weighting function is given by $\rho_i(\tilde{\psi})$, where $\tilde{\psi}$ is the scheduling variable. The dynamics of the blended system, about the operating point $(\tilde{x}_o,u_o)$, are now considered. The velocity-based linearisation form of (5), at $(\tilde{x}_o,u_o)$, is simply obtained by *freezing* the validity function $\rho_i(\tilde{\psi})$ at the operating point. Hence, the following *linear* system is obtained:

$$\begin{aligned}\ddot{\tilde{x}} &= \left(\sum_i A_i(\tilde{x}_i,u_i)\rho_i(\tilde{\psi}_o)\right)\dot{\tilde{x}} \\ &+ \left(\sum_i B_i(\tilde{x}_i,u_i)\rho_i(\tilde{\psi}_o)\right)\dot{u}\end{aligned} \tag{7}$$

It can be seen that, with the appropriate initial conditions, the solution to (7) is initially tangential to the solution of the velocity-based multiple model system (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 (7) as:

$$\ddot{\tilde{x}} = \sum_i \rho_i(\tilde{\psi}_o)\left(A_i(\tilde{x}_i,u_i)\dot{\tilde{x}} + B_i(\tilde{x}_i,u_i)\dot{u}\right) \tag{8}$$

then clearly highlights this direct relationship between the frozen-form, (7), of the velocity-based blended system and the underlying local models, (6), at $(\tilde{x}_o,u_o)$. 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 theoretical analysis reveals that the solution to the frozen form of the velocity-based blended network can be described by the weighted linear combination of the solutions to the local models, giving the properties desirable of a local sub-model framework. Detailed analysis of both the conventional and velocity-based nonlinear representations can be found in (Leith and Leithead, 1999, McLoone, 2000).

3. IMPLEMENTATION OF VELOCITY-BASED LM SYSTEM

In the velocity-based LM network the global dynamics are obtained by blending the local model dynamics. This is not the same as weighting the outputs of each local model to give the global network output, as was originally envisaged for local modelling. Indeed for a heterogeneous LM network (Murray-Smith, 1994), where local models can vary from being complex neural networks to simple constants within one network, weighting the local model outputs is the most obvious, and possibly only, choice for blending. However, if the velocity-based local models are blended in this way the results are rather different from the velocity-based structure proposed by Leith and Leithead (1999). In this regard the possible implementations of the velocity-based network are now considered.

3.1 Local model realisation

Consider the following velocity-based, local state-space model:

$$\begin{aligned} \ddot{\bar{x}} &= A\dot{\bar{x}} + B\dot{u} \\ \dot{\bar{y}} &= C\dot{\bar{x}} \end{aligned} \tag{9}$$

where A, B and C are constant matrices and/or vectors. This local model can be implemented as illustrated in Fig. 1.

When a velocity-based network contains only one local model the global output is obtained by integrating the output of that model. Since no blending of models takes place the corresponding conventional affine local model network produces the exact same output under the appropriate initial conditions. Implementation of blended multiple model networks however is not as straightforward.

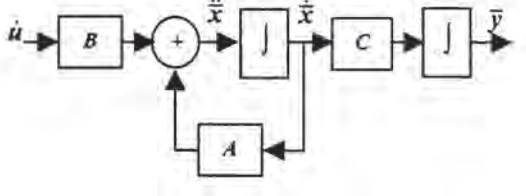


Fig. 1. Velocity-based local model implementation

3.2 Multiple model realisation

For simplicity consider now a velocity-based multiple model network constructed from the blending of two local models. Three different implementations of the nonlinear network are possible since the local models can be blended in three different ways, as illustrated in Fig. 2, 3 and 4 respectively. The results in section 5 will reveal that only the third one is actually correct. In each figure, ρ represents the weighting function.

The first implementation in Fig. 2 is based on weighting the output of each local model, giving the global output:

$$\tilde{y} = \sum_{i=1}^{2} \rho_i \bar{y}_i = C \sum_{i=1}^{2} \rho_i \bar{x}_i \tag{10}$$

It is worth noting that this network implementation employs local state derivative feedback. Global feedback is also possible by blending the local state derivatives, using the same weighting functions, and returning the global state derivative to each local model.

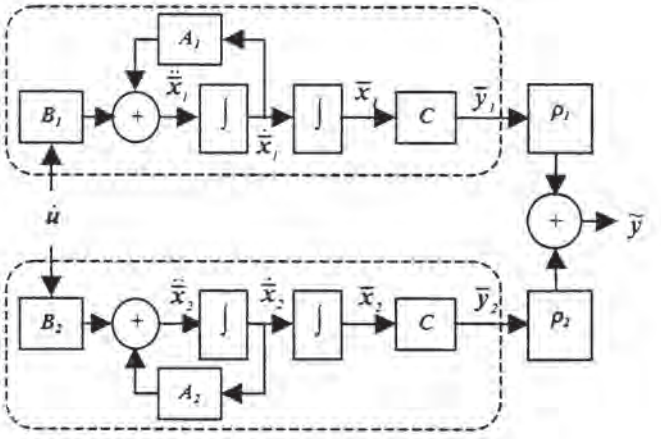


Fig. 2. Velocity-based multiple model implementation – blending $\bar{y}$ (or $\bar{x}$)

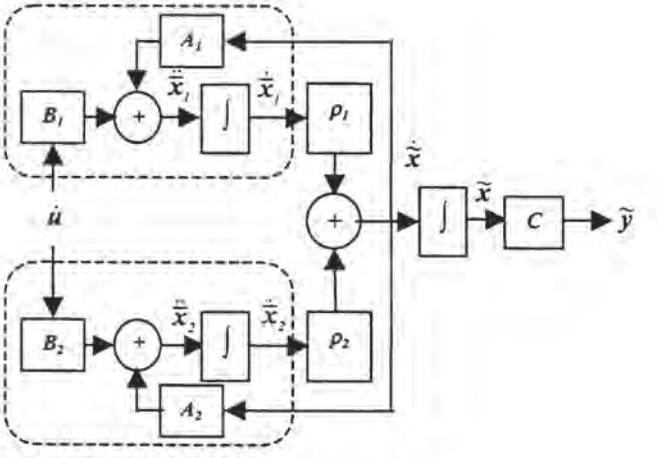


Fig. 3. Velocity-based multiple model implementation – blending $\dot{\tilde{x}}$

The second implementation consists of combining the local model state derivatives, $\dot{\tilde{x}}_i$. The network output is then obtained by integrating the global state derivative:

$$\tilde{y} = C\sum_{i=1}^{2}\int \rho_i \dot{\tilde{x}}_i \qquad (11)$$

Once again, global or local state derivative feedback can be used. Fig. 3 shows the network implementation with global feedback.

The third and final implementation in Fig. 4 involves weighting second derivatives $\ddot{\tilde{x}}_i$ of the local model state to give the global output:

$$\tilde{y} = C\sum_{i=1}^{2}\iint \rho_i \ddot{\tilde{x}}_i \qquad (12)$$

In this case there is no difference between global and local state derivative feedback.

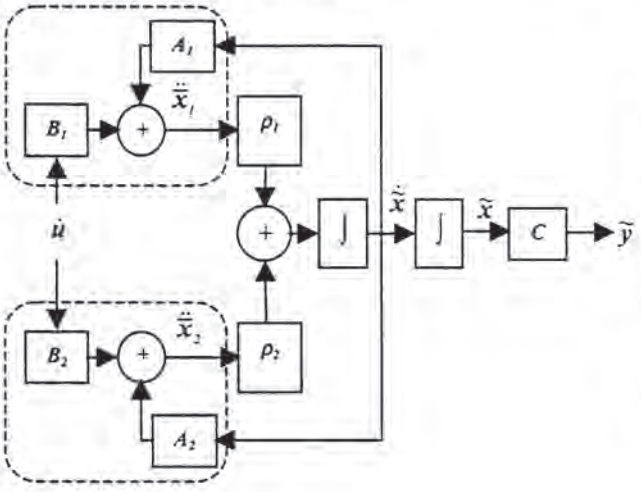


Fig. 4. Velocity-based multiple model implementation – blending $\ddot{\tilde{x}}$

The modelling capabilities of these alternative nonlinear representations are now examined using the CSTR process, which is briefly described in the next section.

4. THE CSTR PROCESS

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

Table 1 Parameters for the simulated CSTR process

q_f	product flowrate	*100* l/min
C_f	input product conc.	*1* mol/l
T_f	input temperature	*350* K
V	container volume	*100* l
E/R	activation energy	10^4 K
T_{cf}	temp. of coolant	*350* K
K_0	plant constant	7.2×10^{10} /min
K_1	plant constant	1.44×10^{13} Kl/min/mol
K_2	plant constant	*0.01* /l
K_3	plant constant	*700* l/min

The CSTR plant can be represented by the following equations:

$$\dot{T}(t) = \frac{q_f}{V}(T_f - T(t)) + K_1 C(t)\exp\left(-\frac{E}{RT(t)}\right) + K_2 q_c(t)\left(1 - \exp\left(-\frac{K_3}{q_c(t)}\right)\right)(T_{cf} - T(t)) \qquad (13)$$

$$\dot{C}(t) = \frac{q_f}{V}(C_f - C(t)) - K_0 C(t)\exp\left(-\frac{E}{RT(t)}\right)$$

where $q_c(t)$ is the input coolant flowrate, the internal plant state, $T(t)$, is the temperature of the solution and $C(t)$ is the output product concentration. The other parameters are their values are given in table 1.

The CSTR plant equations contain nonlinear product and exponential terms. Furthermore, the process dynamics vary considerably across the input space providing a suitably challenging and useful nonlinear system on which to examine the modelling capabilities of the various nonlinear velocity-based representations outlined in the previous section.

5. MODELLING RESULTS AND ANALYSIS

It is the aim of this paper to highlight the need for correct blending of the sub-models within the velocity-based LM framework. Thus, networks with two local models were employed to model the nonlinear dynamical relationship between the input coolant flowrate, q_c, and the output product concentration, C, within the operating space bounded by the input q_c = *[85 111]* l/min.

A nonlinear plant representation was formed by weighting together several suitably located velocity-based linear local models. These local models were obtained by freezing the nonlinear velocity-form of equation (13) at the appropriate linearisation points, where the linearisation point for the i^{th} local model is given by (C_o^i, T_o^i, q_{co}^i).

The equilibrium linearisation points for the models were *(0.0620* mol/l, *448.7522* K, *90* l/min*)* and *(0.1298* mol/l, *432.9487* K, *110* l/min*)*. The two models were blended together using normalised

Gaussian weighting functions. For simplicity only the output C was used as a scheduling variable. The centres of the un-normalised Gaussians were manually chosen to be *0.0620* mol/l and *0.1298* mol/l, with widths of *0.02* and *0.03* respectively. While these weighting functions are not necessarily optimal, they are more than adequate for the purposes of this study.

Implementation of the velocity-based networks requires the derivative of the input q_c to be available. This is difficult to achieve in practice, especially if step changes are involved. Here, a combination of a sinusoid and a constant was used to provide a suitable approximation to a step input. Reducing the sinusoid frequency results in a more accurate step change representation. With this technique the input derivative was accurately implemented and allowed the plant dynamics to be suitably excited for modelling purposes.

Four velocity-based networks were developed. In the first one the LM outputs were blended. The second and third networks both blended the LM state derivatives, but one employed local state feedback while the other used global state feedback. The final network weighted the LM second-order state derivatives.

Simulation results: A series of approximate step changes in q_c, shown in Fig. 5, were applied to each of the above nonlinear systems. The resulting product concentration in each case is displayed in Fig. 6.

These graphs show relatively poor modelling results for all the velocity-based LM networks, especially in terms of the steady-state errors. These occur because of a modelling error inherent in the velocity-based approach. Detail regarding this error can be found in (McLoone, 2000, McLoone and Irwin, 2001). This error cannot be removed by the network and will therefore accumulate over time to produce the poor steady-state results shown in Fig. 6. However, in control applications integral action can easily be used to eliminate the effect of this modelling error. The dynamic performance of the network is therefore more important.

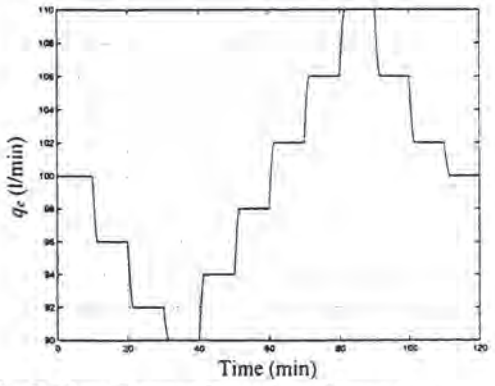


Fig. 5. Step changes in coolant flowrate, q_c

The results show that only the network using weighted second-order state derivatives, (12), reflects the dynamics of the CSTR model. All the other models produce inaccurate dynamics. For example, consider the operating space about the product concentration *0.095* mol/l. Here, the dynamics of the plant are slightly underdamped. The output of the network in Fig. 6(d) reflects this reasonably accurately whereas all the other networks produce significantly more underdamped responses. Note that the network using weighted LM state derivatives, (11), produces a different output when local feedback is used instead of global feedback. However neither reflect the correct dynamics of the CSTR model.

The difference between the various network representations become more apparent when equations (10)-(12) are rewritten in terms of the global second-order state derivative. Thus, equation (10) can be rearranged as follows:

$$\tilde{x} = \sum_i \rho_i \bar{x}_i \Rightarrow \ddot{\tilde{x}} = \frac{\partial^2 \left(\sum_i \rho_i \bar{x}_i \right)}{\partial t^2} \tag{14}$$

$$\Rightarrow \ddot{\tilde{x}} = \sum_i \rho_i \ddot{\bar{x}}_i + 2\sum_i \dot{\rho}_i \dot{\bar{x}}_i + \sum_i \ddot{\rho}_i \bar{x}_i$$

Similarly equation (11) is rewritten as:

$$\ddot{\tilde{x}} = \sum_i \rho_i \ddot{\bar{x}}_i + \sum_i \dot{\rho}_i \dot{\bar{x}}_i \tag{15}$$

and equation (12) becomes:

$$\ddot{\tilde{x}} = \sum_i \rho_i \ddot{\bar{x}}_i \tag{16}$$

When the weighting function, ρ_i, is a constant value, $\dot{\rho}_i = 0$ and all three implementations give the same outputs assuming appropriate initial conditions are used as well as global feedback employed. However when local models are blended in LM networks the weighting functions can vary with time and thus $\dot{\rho}_i \neq 0$. In this case the above implementations will produce significantly different results.

The modelling and control of a nonlinear plant requires accurate representation of the plant dynamics. Local models linearised at particular operating points will accurately model the dynamics of the plant around those points. Intuitively, the dynamics between operating points should simply be a weighting of the local model dynamics. In order to achieve this, velocity-based local models were required. The second-order state derivative, $\ddot{\bar{x}}$, reflects the dynamics of the velocity-based local model. Hence the global dynamics of the velocity-based LM network should be a blended combination of $\ddot{\bar{x}}$. The LM network representation given by (12) clearly achieves this desired blending. All the other representations contain unnecessary additional terms that relate to the rate of change of the weighting

functions. If the rate of change is slow then these terms can be neglected. When the rate of change is not slow these extra terms can contribute significantly to the network output thus detracting from the desired response.

6. CONCLUDING DISCUSSION

The choice of local models and the way in which the models are blended are very important in the local modelling process. In order to achieve accurate plant dynamics using the local modelling approach the LM network should consist of velocity-based local models whose local dynamics are blended together to give the overall global model dynamics.

The actual weighting functions also have an important part to play in the local modelling process. Here, the commonly-chosen normalised Gaussian basis functions were employed. However, alternative weighting functions are possible. Do Gaussians provide the best choice for blending? Future work involves investigating this question in more detail.

REFERENCES

Brown, M. D., Lightbody, G. & Irwin, G. W. (1997). Local model networks for non-linear internal model control. IEE Proc.-D, 144(6), 505-514.

Johansen, T. A. & Foss, B. A. (1992). A NARMAX model representation for adaptive control based on local models. Modelling, Identification and Control 13, 25-39.

Johansen, T. A. & Foss, B. A. (1993). Constructing NARMAX models using ARMAX models. Int. J. Control 58, 1125-1153.

Leith, D. J. & Leithead, W. E. (1999). Analytic framework for blended multiple model systems using linear local models. International Journal of Control, 72, (7/8), 605-619.

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

McLoone, S. C. & Irwin, G. W. (1998). Process dynamic modelling using continuous-time local model networks. Proc. 5th IFAC Workshop on Algorithms and Architectures for real-time control, Mexico, 179-184.

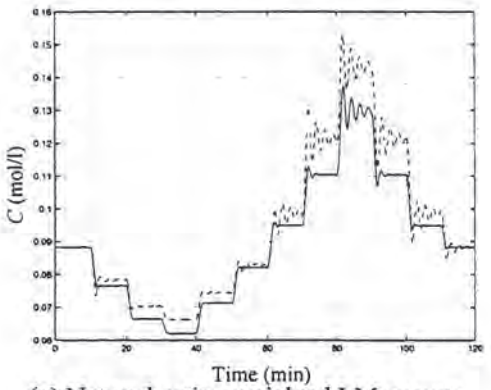
McLoone, S. C. & Irwin, G. W. (2001). On nonlinear modelling using velocity-based multiple model networks. ACC01, Arlington, USA, June 25-27, 2001.

Murray-Smith, R. (1994). A local model network approach to nonlinear modelling. Ph.D. Thesis, University of Strathclyde, UK.

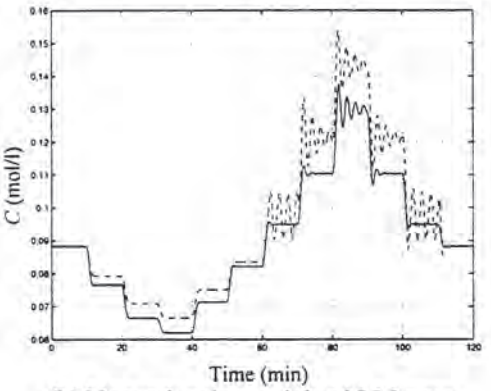
Murray-Smith, R., Johansen, T. A. & Shorten, R. (1999). On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures. ECC'99, Karlsruhe, Germany, F1020-7.

Shorten, R., Murray-Smith, R., Bjorgan, R. & Gollee, H. (1999). On the interpretation of local models in blended multiple model structures. International Journal of Control, 72, (7/8), 620-628.

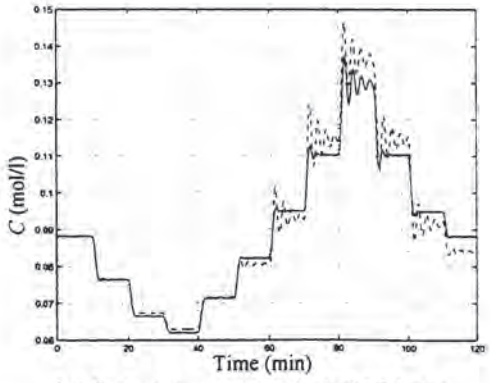
Townsend, S. & Irwin, G. W. (1999). Local model networks for nonlinear predictive control. EUFIT'99, 7th European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, Sept. 13-16.



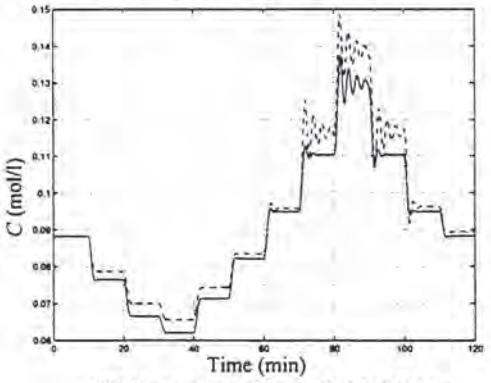
(a) Network using weighted LM outputs



(b) Network using weighted LM state derivatives with global feedback



(c) Network using weighted LM state derivatives with local feedback



(d) Network using weighted LM second-order state derivatives

Fig. 6 Output product concentration, C – in all cases, the solid line represents the actual CSTR plant output, while the dashed line represents the output of the respective network representations