



Nonlinear Model Predictive Control of a Simulated Multivariable Polymerization Reactor Using Second-order Volterra Models

BRYON R. MANER,[†] FRANCIS J. DOYLE III,[†] BABATUNDE A. OGUNNAIKE[‡]
and RONALD K. PEARSON[‡]

An MPC algorithm based on the second-order Volterra model is proposed, and a method for obtaining the model parameters from fundamental models is outlined. Simulation results for two case studies are presented.

Key Words—Nonlinear control systems; process control; process models; predictive control; inverse systems.

Abstract—Two formulations of a nonlinear model predictive control scheme based on the second-order Volterra series model are presented. The first formulation determines the control action using successive substitution, and the second method directly solves a fourth-order nonlinear programming problem on-line. One case study is presented for the SISO control of an isothermal reactor which utilizes the first controller formulation. A second case study is presented for the multivariable control of a large reactor, and uses the nonlinear programming formulation for the controller. The model coefficients for both examples are obtained by discretizing the bilinear Taylor series approximation of the fundamental model and calculating Markov parameters. The relationships between discrete and continuous-time bilinear model matrices using an explicit fourth-order Runge–Kutta method are also included. The responses to set-point changes of both reactors controlled with a linear model predictive control scheme and the second-order Volterra model predictive control scheme are compared to desired, linear reference trajectories. In the majority of the cases examined, the responses obtained by the Volterra controller followed the reference trajectories more closely. Practical issues, including the reduction of the number of model parameters, are addressed in both case studies. Copyright ©1996 Elsevier Science Ltd.

1. INTRODUCTION

Model predictive control (MPC) involves the calculation of a set of manipulated variable moves to minimize an open-loop performance objective over a prediction horizon subject to constraints on

manipulated and controlled variables. Most MPC applications are based on linear finite convolution models, such as the step and impulse response models. An attractive feature of these models is the relative ease with which they can be obtained from plant data. Hence, these models may be obtained in a relatively short time without detailed process knowledge.

Several researchers have proposed predictive control schemes based on the direct use of nonlinear models. In nonlinear quadratic dynamic matrix control (García, 1984), a nonlinear model is used to compute the effect of past manipulated variables on the predicted output. A linear model, obtained by linearizing the nonlinear model at each sampling time, is used to compute the manipulated variable values. An advantage of this approach is that only one quadratic program is solved at each sampling interval versus solving a higher-order nonlinear program. Gattu and Zafriou (1992) extended this algorithm by incorporating state estimation using a steady-state Kalman filter gain computed at each sampling time. The performance of the algorithm was illustrated in the control of a two-by-two continuous stirred tank reactor (CSTR) and a single-input–single-output open-loop unstable reactor. Lee and Ricker (1993) modified García's algorithm by implementing an extended Kalman filter and by incorporating load disturbances on the states as additive stochastic signals. In Gattu and Zafriou (1992), the Kalman filter was designed with white state noise characterized by a scalar-times-identity covariance matrix, which can lead to significant bias in the state estimates. Ricker and Lee implemented their algorithm in an eight-by-eight nonlinear MPC scheme for the Tennessee Eastman test problem (Ricker and Lee, 1995).

Received 15 August 1995; revised 20 March 1996; received in final form 5 May 1996. This paper was not presented at any IFAC meeting. This paper was recommended in revised form by Associate Editor B. Wayne Bequette under the direction of Editor Yaman Arkun. Corresponding author Professor Frances J. Doyle III. Tel. +1 317 494 9472; Fax +1 317 494 0805; E-mail fdoyle@ecn.purdue.edu.

[†] School of Chemical Engineering, Purdue University, West Lafayette, IN 47907-1283, U.S.A.

[‡] E.I. DuPont de Nemours & Co., Wilmington, DE 19880-0101, U.S.A.

Li and Biegler (1988) extended the nonlinear internal model control design procedure (Economou and Morari, 1985) to a single-step method that included constraints on the states and inputs. They demonstrated the performance of their controller for the control of two stirred tank reactors with two inputs and two outputs. The stability of the constrained controller was guaranteed by the small gain theorem (Zames, 1966) provided that the process was open-loop stable. They extended their algorithm to a multi-step strategy that linearizes a nonlinear model around a nominal trajectory (Li and Biegler, 1989). The modified algorithm was demonstrated with the optimal open-loop control of an SISO and an MIMO (3×3) CSTR. In the case studies, it was assumed that there was no plant-model mismatch.

Both the nonlinear quadratic dynamic matrix control (Garcia, 1984; Gattu and Zafiriou, 1992; Lee and Ricker, 1993) and modified nonlinear internal model control (Li and Biegler, 1988; Li and Biegler, 1989) algorithms require a fundamental model of the process. The derivation of these models can be very time consuming, and can be elusive if the process is not well understood. One nonlinear MPC approach that does not require the availability of a fundamental model uses polynomial ARMA models obtained from input-output data (Hernández, 1992). The main advantage of using polynomial models is that the one step ahead prediction problem can be formulated as a linear regression, greatly simplifying the estimation of the parameters from input-output data.

This paper presents a natural extension of the linear MPC algorithm described in (Ricker, 1985) to a control methodology based on the second-order Volterra series model. The control action can be computed via successive substitution or through the direct solution of a fourth-order nonlinear program. A second-order Volterra model can describe nonlinear behavior such as asymmetric output changes in response to symmetric changes in the input. A controller based on this model can yield improved performance over a linear model-based controller. In addition, the model parameters can be obtained from both Carleman linearization of a nonlinear fundamental model (Rugh, 1981), and input-output data (Pearson *et al.*, 1996). Identification of these parameters from input-output data enables implementation of this nonlinear MPC scheme in the absence of a fundamental model. Another advantage of a second-order Volterra model-based controller is that it can be posed as a fourth-order nonlinear program. While this nonlinear program is more computationally demanding than the quadratic program that arises in linear MPC, it is more readily solvable than the general

nonlinear programs that arise in the nonlinear internal model control algorithm and nonlinear MPC based on polynomial ARMA models noted above. In particular, one group of researchers (Chow *et al.*, 1994) has developed an algorithm for unconstrained optimization using tensor methods that employs a fourth-order model of the objective function. In their case studies, the tensor method required significantly fewer iterations and function evaluations to solve most unconstrained optimization problems than standard methods based on quadratic models. A third advantage is that stability results are easier to obtain for Volterra model-based controllers, since the model depends only on past inputs versus the dependence on both past inputs and outputs as in the case for polynomial ARMA model-based controllers. Stability results for second-order Volterra model-based controllers have been presented by several researchers (Genceli and Nikolaou, 1995; Zheng and Zafiriou, 1993).

2. CONTROLLER FORMULATIONS

2.1. Successive substitution

The triangular form of the second-order Volterra model with truncation order N is given by the following expression:

$$y(k) = y_0 + \sum_{i=1}^N a_i u(k-i) + \sum_{i=1}^N \sum_{j=i}^N b_{i,j} u(k-i) u(k-j). \quad (1)$$

The triangular form may be used without loss of generality because the second-order parameters are symmetric for the Volterra model. The y_0 term is a bias term that appears in the general form of the second-order Volterra model. The a_i terms are impulse response coefficients, and the $b_{i,j}$ terms are the second-order Volterra model parameters. The first and second terms on the right-hand side of equation (1) are first and second-order terms which will be denoted as $\tilde{P}_1[u]$ and $\tilde{P}_2[u]$, respectively. The block diagram for the second-order Volterra model predictive controller is shown in Fig. 1. P is the actual nonlinear plant being controlled. \tilde{P}_1^\dagger is an inverse or pseudo-inverse of the linear model and is identical to the inverse used in linear MPC; ξ is the difference between a vector of desired future setpoints and a vector of future predicted outputs due to past inputs. It can be seen from Fig. 1 that the second-order Volterra model-based algorithm is an extension of the constrained internal model control algorithm described by Ricker (1985). The second-order Volterra model replaces the impulse response model and is used for output prediction.

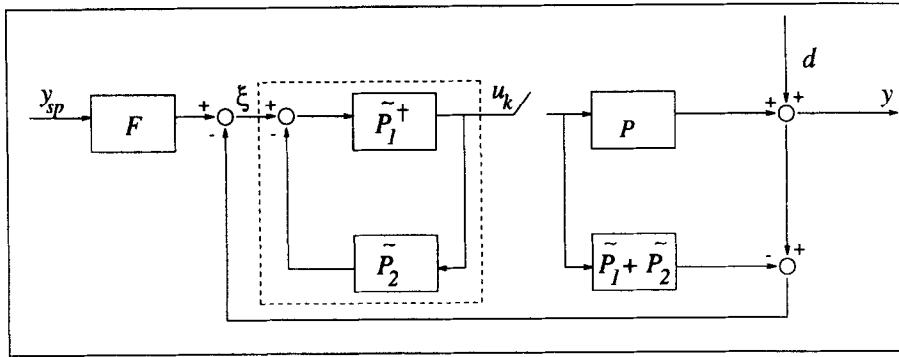


Fig. 1. Block diagram for the second-order Volterra model predictive controller.

The corresponding Volterra controller, enclosed by the dashed box in Fig. 1, replaces the linear model-based controller and performs the control action computations. For the constrained case, a constrained quadratic program is solved at each iteration through the $\tilde{P}_1^+ - \tilde{P}_2$ loop in Fig. 1. Hence, the iterative loop guarantees that the control action calculated by the Volterra controller satisfies the constraints on the manipulated and controlled variables. The model equations for the SISO case are given below

$$\begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+P) \end{bmatrix} = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_1 & \ddots & 0 \\ \vdots & \vdots & \ddots & a_1 \\ \vdots & \vdots & \ddots & a_1 + a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_P & a_{P-1} & \cdots & \sum_{i=1}^{P-M+1} a_i \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+M-1) \end{bmatrix} + \begin{bmatrix} c(k+1) \\ c(k+2) \\ \vdots \\ c(k+P) \end{bmatrix} + \begin{bmatrix} f(k+1) \\ f(k+2) \\ \vdots \\ f(k+P) \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} c(k+1) \\ c(k+2) \\ \vdots \\ c(k+P) \end{bmatrix} = \begin{bmatrix} a_2 & a_3 & \cdots & a_N & 0 \\ a_3 & a_4 & \cdots & a_{N-1} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a_{P-1} & a_P & \vdots & \vdots & \vdots & \vdots \\ a_P & 0 & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \vdots & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u(k-1) \\ u(k-2) \\ \vdots \\ u(k-N) \end{bmatrix} + \begin{bmatrix} d(k+1) \\ d(k+1) \\ \vdots \\ d(k+1) \end{bmatrix} + \begin{bmatrix} g(k+1) \\ g(k+2) \\ \vdots \\ g(k+P) \end{bmatrix} \quad (3)$$

P , M , and N are the output prediction horizon, input move horizon, and model truncation order, respectively. The future-future and future-past cross terms of the input correspond to $f(k+1)$, $f(k+2)$, ..., etc.

The past-past cross terms of the input appear as $g(k+1)$, $g(k+2)$, ..., etc. $d(k+1)$ is the difference between the measured value of the process output and predicted model output at time k . Introducing vector notation, equations (2) and (3) may be written in a more compact form as shown below

$$y = Gu + c + f, \quad (4)$$

$$c = Hu_{\text{past}} + d + g. \quad (5)$$

Equations (4) and (5) are similar to the equations in Ricker (1985) except for the presence of the f and g terms which arise due to the second-order model. c is the invariant contribution of past inputs and does not change with iterations around the $\tilde{P}_1^+ - \tilde{P}_2$ loop in Fig. 1. G and H are matrices that contain impulse response coefficients. u_{past} is an historical record of past values of the input, and u is the linear contribution of the present and future values of the input. In addition, a setpoint trajectory, s , may be implemented via the following equations:

$$s(k+1) = \lambda y_{\text{meas}} + (1-\lambda)y_{\text{sp}}, \quad (6)$$

$$s(k+i) = \lambda s(k+i-1) + (1-\lambda)y_{\text{sp}}, \quad 2 \leq i \leq P, \quad (7)$$

where λ is a tuning parameter (in addition to M and P) used to filter the setpoint.

To illustrate the computation of f and g , consider the SISO case where $M = 2$, $P = 3$, and $N = 4$. Define B , a matrix containing the second-order coefficients, as

$$B = \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} \\ 0 & b_{2,2} & b_{2,3} & b_{2,4} \\ 0 & 0 & b_{3,3} & b_{3,4} \\ 0 & 0 & 0 & b_{4,4} \end{bmatrix} \quad (8)$$

The $b_{i,j}$ terms represent coefficients in a triangular Volterra model (equation (1)). At time k , $u(k)$ and $u(k+1)$ are calculated. The entries in f are given by

$$f(k+1) = [u(k) \ 0 \ 0 \ 0]B$$

$$\times [u(k) \ u(k-1) \ u(k-2) \ u(k-3)]^T, \quad (9)$$

$$f(k+2) = [u(k+1) \ u(k) \ 0 \ 0] \mathbf{B} \\ \times [u(k+1) \ u(k) \ u(k-1) \ u(k-2)]^T, \quad (10)$$

$$f(k+3) = [u(k+2) \ u(k+1) \ u(k) \ 0] \mathbf{B} \\ \times [u(k+2) \ u(k+1) \ u(k) \ u(k-1)]^T. \quad (11)$$

Since $M = 2$, $u(k+1) = u(k+2)$ in equation (11). The entries in \mathbf{g} are given by

$$\mathbf{g}(k+1) = [0 \ u(k-1) \ u(k-2) \ u(k-3)] \mathbf{B} \\ \times [0 \ u(k-1) \ u(k-2) \ u(k-3)]^T, \quad (12)$$

$$\mathbf{g}(k+2) = [0 \ 0 \ u(k-1) \ u(k-2)] \mathbf{B} \\ \times [0 \ 0 \ u(k-1) \ u(k-2)]^T, \quad (13)$$

$$\mathbf{g}(k+3) = [0 \ 0 \ 0 \ u(k-1)] \mathbf{B} \\ \times [0 \ 0 \ 0 \ u(k-1)]^T. \quad (14)$$

For the unconstrained case, the control action is computed at time k using the following algorithm:

Step 1. Set $i = 1$.

Step 2. Calculate \mathbf{a} and \mathbf{u} by solving the unconstrained least squares control problem

$$\mathbf{a} = ((\mathbf{s} - \mathbf{c} - \mathbf{f})^T \mathbf{G})^T, \quad (15)$$

$$\mathbf{u} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{a}. \quad (16)$$

Step 3. Determine if the condition in equation (17) is satisfied where δ is the desired tolerance

$$|u^{(i)}(k) - u^{(i-1)}(k)| \leq \delta \quad (17)$$

and the superscript denotes the iteration step.

Step 4. • If yes, set $u(k) = u^{(i)}(k)$. Close the switch in Fig. 1 and implement $u(k)$.

- If no, recalculate \mathbf{f} using $u^{(i)}(k)$ for present and future values of the input. Set $i = i + 1$. Return to Step 2. (For the constrained case, \mathbf{u} is calculated by solving a quadratic program rather than using the least-squares solution in equation (16).)

Contraction mapping arguments can be made to show convergence of the successive substitution algorithm (Economou, 1985). Additional details of the algorithm as well as examples of the constraint handling and disturbance rejection capabilities appear in Maner (1993).

2.2. Nonlinear programming formulation

The nonlinear programming formulation requires solving the following optimization problem at each sampling interval:

$$\min_{\mathbf{u}} \sum_{i=1}^P y_i (s(k+i) - y(k+i))^2 \\ + \sum_{j=1}^M \lambda_j (\Delta u(k+j-1))^2$$

subject to

$$y(k+i) = \sum_{\ell=1}^N a_{\ell} u(k+i-\ell) \\ + \sum_{\ell=1}^N \sum_{m=\ell}^N b_{\ell,m} u(k+i-\ell) u(k+i-m), \\ \mathbf{u} = [u(k) \ u(k+1) \ \dots \ u(k+M-1)]^T, \quad (18)$$

$$\mathbf{u}_{\text{low}} \leq \mathbf{u} \leq \mathbf{u}_{\text{high}}, \\ \Delta \mathbf{u}_{\text{low}} \leq \Delta \mathbf{u} \leq \Delta \mathbf{u}_{\text{high}}.$$

The approach in Section 2.1 involves a successive substitution approach to solve the optimization problem in equation (18). An alternative method to calculate the control action would employ a general nonlinear programming solver to directly solve the optimization problem. It should be noted that equation (18) does not include output constraints. Their inclusion would lead to a more complex nonlinear programming problem involving nonlinear constraints as well as a nonlinear objective function.

3. CALCULATION OF MODEL PARAMETERS FOR A MULTIVARIABLE CASE

The multivariable equation for output i having q inputs for the second-order Volterra model used in this work is given by

$$y_i(k) = y_{i0} + \sum_{l=1}^q \sum_{j=1}^N a_{l,j}^i u_l(k-j) \\ + \sum_{l=1}^q \sum_{j=1}^N \sum_{n=j}^N b_{l,j,l,n}^i u_l(k-j) u_l(k-n), \quad (19)$$

i.e. no cross terms in the inputs were permitted. The y_{i0} term in equation (19) is a bias term whose value could be fitted using input-output data. However, no bias term arises when an analytical construction of the model parameters is used. To illustrate the procedure for the multivariable case, a two-input-two-output system will be considered. A

continuous-time bilinear model is obtained from a fundamental model using Carleman linearization (Rugh, 1981)

$$\dot{\mathbf{z}}(t) = \mathbf{A}\mathbf{z}(t) + \mathbf{N}_1\mathbf{z}(t)u_1(t) + \mathbf{N}_2\mathbf{z}(t)u_2(t) + \mathbf{B}\mathbf{u}(t) \equiv \mathbf{f}(\mathbf{z}(t), \mathbf{u}(t)), \quad (20)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{z}(t). \quad (21)$$

Equations (20) and (21) describe the resulting bilinear model where the variables, \mathbf{u} , \mathbf{x} , and \mathbf{y} are in deviation form, and the state vector, \mathbf{x} , contains the original states (x_1, x_2, \dots) and the second-order cross terms of the states (x_1^2, x_1x_2, \dots). The inputs and outputs are scaled to produce $a_{l,j}^i$ and $b_{l,j,l,n}^i$ parameters that yield well-conditioned matrices for use in nonlinear MPC. Poor scaling can lead to singularity and roundoff problems which would be problematic for a nonlinear programming solver. Diagonal scaling matrices were introduced for the inputs and outputs

$$\mathbf{N}_u = \begin{bmatrix} N_{u_1} & 0 \\ 0 & N_{u_2} \end{bmatrix}, \quad (22)$$

$$\mathbf{N}_y = \begin{bmatrix} N_{y_1} & 0 \\ 0 & N_{y_2} \end{bmatrix} \quad (23)$$

and the state space matrices were redefined as follows:

$$\hat{\mathbf{N}}_i = \mathbf{N}_i\mathbf{N}_{u_i}, \quad i = 1, 2, \quad (24)$$

$$\hat{\mathbf{B}} = \mathbf{B}\mathbf{N}_u, \quad (25)$$

$$\hat{\mathbf{C}} = \mathbf{N}_y^{-1}\mathbf{C}. \quad (26)$$

In this work, the scaling factors were chosen to be approximately equal to the maximum range of each input and output.

A fourth-order explicit Runge-Kutta scheme is used to obtain a discrete-time bilinear model. Since the full nonlinear equations are integrated over a time interval while holding the manipulated variables constant, \mathbf{u} was treated as a constant vector in the derivation of the discretization scheme. One fourth-order Runge-Kutta method with discretization step size h uses the following equation (Carnahan *et al.*, 1969):

$$\mathbf{z}(k+1) = \mathbf{z}(k) + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4), \quad (27)$$

where

$$\mathbf{k}_1 = \mathbf{f}(\mathbf{z}(k)), \quad (28)$$

$$\mathbf{k}_2 = \mathbf{f}(\mathbf{z}(k) + \frac{1}{2}h\mathbf{k}_1), \quad (29)$$

$$\mathbf{k}_3 = \mathbf{f}(\mathbf{z}(k) + \frac{1}{2}h\mathbf{k}_2), \quad (30)$$

$$\mathbf{k}_4 = \mathbf{f}(\mathbf{z}(k) + h\mathbf{k}_3). \quad (31)$$

The resulting discrete-time model matrices for the bilinear model are

$$\bar{\mathbf{A}} \equiv \mathbf{I} + \frac{h}{6} \left[\frac{1}{4}h^3\mathbf{A}^4 + h^2\mathbf{A}^3 + 3h\mathbf{A}^2 + 6\mathbf{A} \right], \quad (32)$$

$$\begin{aligned} \bar{\mathbf{N}}_i &\equiv \frac{h}{6} \left[\frac{1}{4}h^3 \left(\mathbf{A}^3\hat{\mathbf{N}}_i + \mathbf{A}^2\hat{\mathbf{N}}_i\mathbf{A} + \mathbf{A}\hat{\mathbf{N}}_i\mathbf{A}^2 + \hat{\mathbf{N}}_i\mathbf{A}^3 \right) \right. \\ &\quad \left. + \frac{h}{6} \left[h^2 \left(\mathbf{A}^2\hat{\mathbf{N}}_i + \mathbf{A}\hat{\mathbf{N}}_i\mathbf{A} + \hat{\mathbf{N}}_i\mathbf{A}^2 \right) \right] \right. \\ &\quad \left. + \frac{h}{6} \left[3h \left(\mathbf{A}\hat{\mathbf{N}}_i + \hat{\mathbf{N}}_i\mathbf{A} \right) + 6\hat{\mathbf{N}}_i \right] \right], \quad i = 1, 2, \end{aligned} \quad (33)$$

$$\bar{\mathbf{B}}^{(1)} \equiv \frac{h}{6} \left[\frac{1}{4}h^3\mathbf{A}^3\hat{\mathbf{B}} + h^2\mathbf{A}^2\hat{\mathbf{B}} + 3h\mathbf{A}\hat{\mathbf{B}} + 6\hat{\mathbf{B}} \right], \quad (34)$$

$$\begin{aligned} \bar{\mathbf{B}}^{(2)}(:, i) &\equiv \frac{h}{6} \left[\frac{1}{4}h^3 \left(\mathbf{A}^2\hat{\mathbf{N}}_i\hat{\mathbf{B}}(:, i) + \mathbf{A}\hat{\mathbf{N}}_i\mathbf{A}\hat{\mathbf{B}}(:, i) + \hat{\mathbf{N}}_i\mathbf{A}^2\hat{\mathbf{B}}(:, i) \right) \right. \\ &\quad \left. + \frac{h}{6} \left[h^2 \left(\mathbf{A}\hat{\mathbf{N}}_i\hat{\mathbf{B}}(:, i) + \hat{\mathbf{N}}_i\mathbf{A}\hat{\mathbf{B}}(:, i) \right) + 3h\hat{\mathbf{N}}_i\hat{\mathbf{B}}(:, i) \right] \right], \\ &\quad i = 1, 2. \end{aligned} \quad (35)$$

The $\bar{\mathbf{B}}^{(2)}$ matrix arises in the derivation of the discretization method. $\hat{\mathbf{B}}(:, i)$ denotes the i th column of the matrix, $\hat{\mathbf{B}}$. The discrete-time model is then given by

$$\begin{aligned} \mathbf{z}(k+1) &= \bar{\mathbf{A}}\mathbf{z}(k) + \bar{\mathbf{N}}_1\mathbf{z}(k)u_1(k) \\ &\quad + \bar{\mathbf{N}}_2\mathbf{z}(k)u_2(k) + \bar{\mathbf{B}}^{(1)}\mathbf{u}(k) \\ &\quad + \bar{\mathbf{B}}^{(2)}(:, 1)u_1^2(k) + \bar{\mathbf{B}}^{(2)}(:, 2)u_2^2(k), \end{aligned} \quad (36)$$

$$\mathbf{y}(k) = \hat{\mathbf{C}}\mathbf{z}(k). \quad (37)$$

A derivation of discrete-time matrices using the improved Euler method and relationships between discrete-time and continuous-time bilinear model matrices are given in the Appendix of Maner (1993). This method yields satisfactory results for bilinear systems that have a low degree of stiffness.

The parameters for a second-order, triangular Volterra model are calculated using the following equations (Rugh, 1981):

$$\begin{bmatrix} a_{1,i}^1 & a_{2,i}^1 \\ a_{1,i}^2 & a_{2,i}^2 \end{bmatrix} = \hat{\mathbf{C}}\bar{\mathbf{A}}^{i-1}\bar{\mathbf{B}}^{(1)}, \quad i \geq 1, \quad (38)$$

$$\begin{bmatrix} b_{1,i,1,i}^1 & b_{2,i,2,i}^1 \\ b_{1,i,1,i}^2 & b_{2,i,2,i}^2 \end{bmatrix} = \hat{\mathbf{C}}\bar{\mathbf{A}}^{i-1}\bar{\mathbf{B}}^{(2)}, \quad i \geq 1, \quad (39)$$

$$\begin{bmatrix} b_{1,j,1,i}^1 \\ b_{1,j,1,i}^2 \end{bmatrix} = \hat{\mathbf{C}}\bar{\mathbf{A}}^{j-1}\bar{\mathbf{N}}_1\bar{\mathbf{A}}^{i-j-1}\bar{\mathbf{B}}^{(1)}(:, 1), \quad j \geq 1, i \geq j+1, \quad (40)$$

$$\begin{bmatrix} b_{2,j,2,i}^1 \\ b_{2,j,2,i}^2 \end{bmatrix} = \hat{\mathbf{C}}\bar{\mathbf{A}}^{j-1}\bar{\mathbf{N}}_2\bar{\mathbf{A}}^{i-j-1}\bar{\mathbf{B}}^{(1)}(:, 2), \quad j \geq 1, i \geq j+1. \quad (41)$$

In equation (38), $a_{l,j}^i$ corresponds to the j th impulse response coefficient relating input l to output i . $b_{l,j,l,n}^i$ corresponds to the second-order Volterra parameter with delays j and n relating input l to output i . The $b_{l,j,m,n}^i$ parameters are not included, since the contribution of these terms did not justify the increased complexity that would be obtained if they were added to the model.

4. SIMULATION RESULTS

4.1. Case Study I: SISO control of an isothermal CSTR

As a first example, consider the isothermal free-radical polymerization of methyl methacrylate using azo-bis-isobutyronitrile as initiator and toluene as solvent (Congalidis *et al.*, 1989; Daoutidis *et al.*, 1990; Doyle III *et al.*, 1995). The number average molecular weight (NAMW) is controlled by manipulating the inlet initiator flow rate. A schematic of the process is shown in Fig. 2. The isothermal model was obtained by setting the reactor temperature equal to its steady-state value of 335 K. Under these assumptions, the six state model in Daoutidis *et al.* (1990) reduces to the following four state model:

$$\frac{dC_m}{dt} = -(k_p + k_{tm})C_mP_0 + \frac{F(C_{m_{in}} - C_m)}{V}, \quad (42)$$

$$\frac{dC_1}{dt} = -k_1C_1 + \frac{F_1C_{1_{in}} - FC_1}{V}, \quad (43)$$

$$\frac{dD_0}{dt} = (0.5k_{T_c} + k_{T_d})P_0^2 + k_{tm}C_mP_0 - \frac{FD_0}{V}, \quad (44)$$

$$\frac{dD_1}{dt} = M_m(k_p + k_{tm})C_mP_0 - \frac{FD_1}{V}, \quad (45)$$

$$y = \frac{D_1}{D_0}. \quad (46)$$

where

$$P_0 = \left[\frac{2f^*k_1C_1}{k_{T_d} + k_{T_c}} \right]^{0.5}.$$

The model parameters and steady-state operating conditions are listed in Tables 1 and 2, respectively. The variables u , x , and y were placed in deviation form and scaled by their nominal values ($\tilde{u} = \frac{u-u_0}{u_0}$, $\tilde{x}_i = \frac{x_i-x_{i0}}{x_{i0}}$, and $\tilde{y} = \frac{y-y_0}{y_0}$), and a bilinear representation of the nonlinear model was obtained using Carleman linearization (Rugh, 1981). The continuous-time bilinear system was then discretized using the explicit fourth-order Runge-Kutta method given by equations (32)–(35) with a sampling time of 0.03 h (1.8 min). The largest negative eigenvalue of the system was -10.9 h^{-1} . The corresponding small time constant for this system arises from the relatively small reactor volume of 100ℓ. The model parameters were calculated using equations (38)–(40). Linear MPC and the Volterra controller described in Section 2.1 were used with $M = 1$, $P = 25$, and $\lambda = 0.95$ as tuning parameters. The tuning parameters were the same for both controllers because the underlying linear models are identical. The truncation order of the model was $N = 30$.

This value was chosen to obtain a model with a memory approximately equal to three times the time constant of the system. The plant was the ac-

Table 1. Parameters for Case Study I

k_{T_c}	=	1.3281×10^{10}	$\text{m}^3/(\text{kmol h})$
k_{T_d}	=	1.0930×10^{11}	$\text{m}^3/(\text{kmol h})$
k_1	=	1.0225×10^{-1}	$1/\text{h}$
k_p	=	2.4952×10^6	$\text{m}^3/(\text{kmol h})$
k_{tm}	=	2.4522×10^3	$\text{m}^3/(\text{kmol h})$
f^*	=	0.58	
F	=	1.00	m^3/h
V	=	0.1	m^3
$C_{1_{in}}$	=	8.0	kmol/m^3
M_m	=	100.12	kg/kmol
$C_{m_{in}}$	=	6.0	kmol/m^3

Table 2. Steady-state operating conditions for Case Study I

$x_1 = C_m$	=	5.506774	kmol/m^3
$x_2 = C_1$	=	0.132906	kmol/m^3
$x_3 = D_0$	=	0.0019752	kmol/m^3
$x_4 = D_1$	=	49.38182	kg/m^3
$u = F_1$	=	0.016783	m^3/h
y	=	25,000.5	kg/kmol

tual nonlinear model integrated with a fourth-order Runge-Kutta method.

Open-loop responses of the nonlinear, linear, and second-order Volterra models to step changes of $\pm 0.008392 \text{ m}^3/\text{h}$ in F_1 from its nominal value of $0.016783 \text{ m}^3/\text{h}$ are shown in Fig. 3. In both simulations, the output of the second-order Volterra model more closely tracks the actual nonlinear plant output. The output of the Volterra model is asymmetric for steps of the same magnitude of opposite sign, while that of the linear model is symmetric.

The result of a closed-loop simulation for a setpoint change from 25,000.5–38,000 kg/kmol is shown in Fig. 4. The reference trajectory was calculated by performing a closed-loop simulation with the same tuning parameters using a linear plant and assuming no plant-model mismatch. The Volterra controller effectively cancels the process nonlinearity to produce a linear response. If all of the nonlinearity was canceled by the controller, the closed-loop response would be identical to the reference trajectory. In this simulation, the response obtained by the Volterra controller tracks the reference trajectory more closely than the response obtained by the linear controller. Linear MPC leads to an overshoot in the setpoint response, and this may be undesirable if product specifications require the molecular weight to be less than 38,000 kg/kmol. The cause of the overshoot can be attributed to the aggressive manipulated variable profile for linear MPC shown in Fig. 4. While linear MPC could be detuned to yield an overdamped response for this setpoint change, detuning would cause performance deterioration for setpoints below the nominal value of 25,000.5 kg/kmol. Hence,

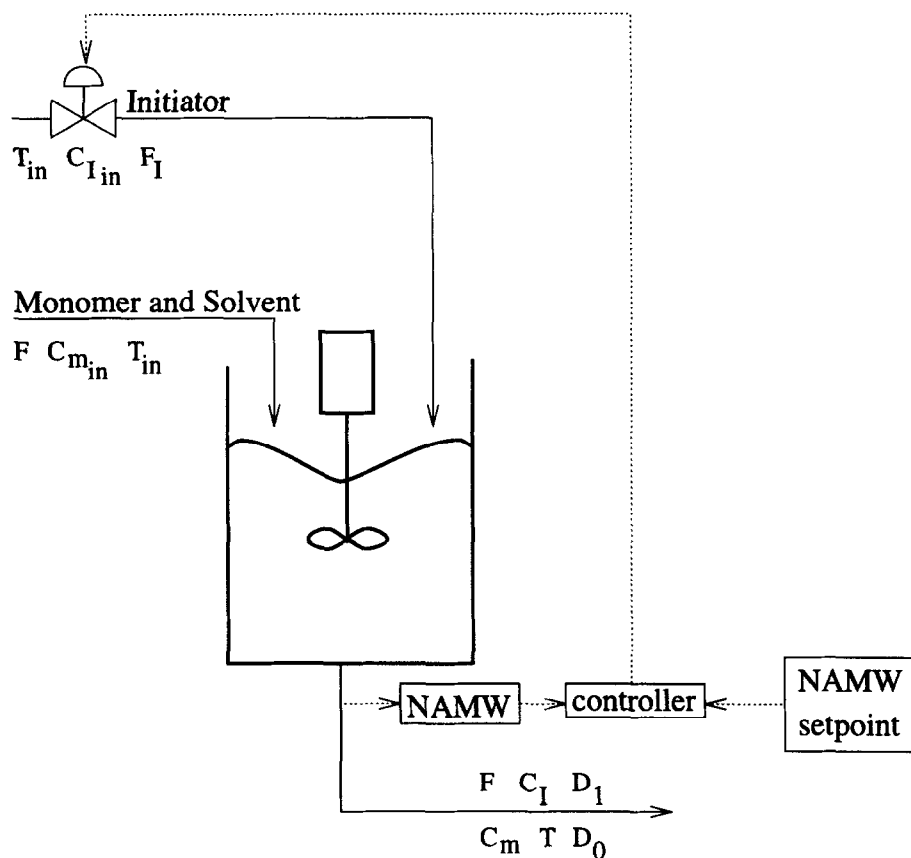
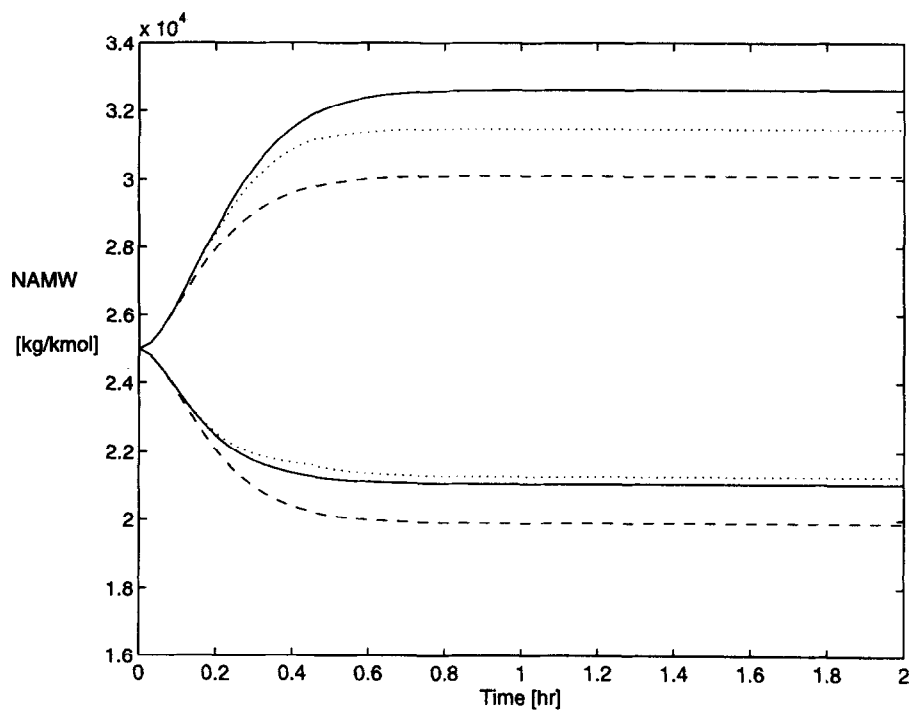


Fig. 2. Control configuration for Case Study I.

Fig. 3. Open-loop simulations for step changes of $\pm 0.008392 \text{ m}^3/\text{h}$ in F_I from its nominal value of $0.016783 \text{ m}^3/\text{h}$. Solid: nonlinear; dashed: linear; dotted: Volterra.

the nonlinear behavior of this process requires a compromise in the tuning of a linear model-based

controller. Results for other setpoint changes are listed in Table 3.

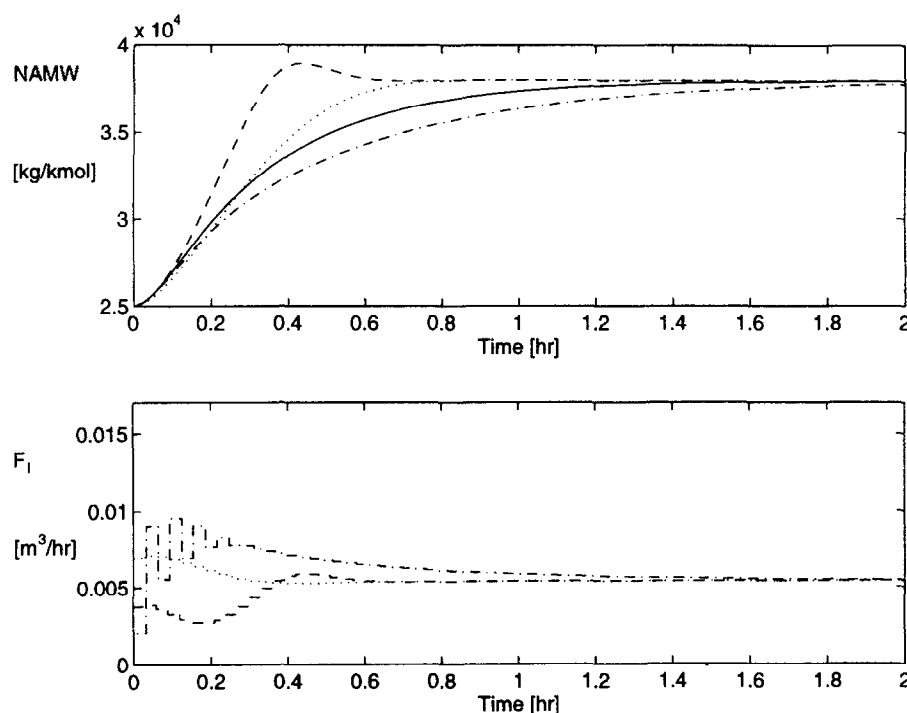


Fig. 4. Closed-loop simulation for a filtered step setpoint change from 25,000.5 to 38,000 kg/kmol. Solid: reference; dashed: linear MPC; dotted: Volterra MPC; dash dot: nonlinear QDMC.

Table 3. Performance comparison of the two controllers for Case Study I

Setpoint kg/kmol	Percent improvement in ISE	max. # of iterations
38,000	84.79	27
32,000	93.23	8
28,000	97.29	4
22,000	74.41	4
18,000	unstable	NA
12,000	unstable	NA

For a comparison with another nonlinear MPC strategy, the dash dot lines in Fig. 4 correspond to input and output profiles obtained using nonlinear quadratic dynamic matrix control (QDMC) (García, 1984) with $M = 1$, $P = 5$, and a first-order filter in the feedback path with filter time constant 0.94. The tuning parameters for nonlinear QDMC were different from those used for linear MPC and Volterra MPC, because the underlying linear models used by nonlinear QDMC were different from those of the other two controllers. The tuning parameters used in nonlinear QDMC resulted in an improvement in closed-loop performance as compared to implementing the tuning parameters used in linear and Volterra MPC. In this implementation, the actual nonlinear model is linearized at each sampling interval, i.e. there is no plant-model mismatch. The output profile is much more sluggish than the output trajectories obtained using linear and Volterra MPC. It might be argued that nonlinear QDMC should be tuned more aggressively.

However, the initial chattering observed in the initiator flow rate using nonlinear QDMC indicates that this controller is already tuned quite aggressively. The poor performance is due to the difficulty that nonlinear QDMC encounters with large input variable changes. Since the linear model employed in linear MPC and the linear model used by nonlinear QDMC at time zero are identical, the first input move is the same for both algorithms. This move corresponds to a change of -87.53% from the nominal input value. The states in a physical nonlinear model would not be expected to change drastically in one sampling period. Hence, the values of the states after one sampling period could be expected to be close to their initial values. Prior to calculating a second control move, nonlinear QDMC linearizes the nonlinear model at the values of the input and states after one sampling interval. However, the linear model can be poor if these values do not correspond to steady-state values. This would occur, for example, if a large input move was computed and the states in the nonlinear model have not had time to respond to this input. This drawback of nonlinear QDMC was noted in the paper presented by García (1984) in that the algorithm approaches the performance of a rigorous nonlinear controller that solves a general nonlinear programming problem "... for small input variable changes and disturbances that do not move the operating point far." For Case Study I, nonlinear QDMC repeatedly obtains poor linear models at the begin-

ning of the simulation as evidenced by the chattering in the initiator flow rate profile.

It is interesting to note the effect of reducing the number of second-order coefficients on the closed-loop performance. Plots of the first and second-order parameters are shown in Figs 5 and 6, respectively. Since the $b_{i,j}$ coefficients are one order of magnitude smaller than the a_i coefficients in equation (1), and \bar{u} is on the order of 10^{-3} , the later $b_{i,j}$ terms in Fig. 6 make a very small contribution to the model. Since $N = 30$, there are 465 $b_{i,j}$ terms. If the truncation order for the second-order parameters is reduced from $N = 30$ –15, there would be 120 $b_{i,j}$ terms. The closed-loop performance using this reduced model is shown in Fig. 7. As expected, the performance is very nearly the same as that shown in Fig. 4 for the full second-order Volterra model. This model structure pruning approach corresponds to reducing the nonlinear memory of the system as noted in Pearson *et al.* (n.d.).

For setpoints below 18,000 kg/kmol, the closed-loop performance of the Volterra controller is unstable while the linear model-based controller successfully brings the nonlinear plant to the new setpoint. An explanation for this behavior can be found from a plot of the steady-state gain loci (Fig. 8). The gain of the bilinear model changes sign in the region of $y = 18,000$ kg/kmol. This error is the result of deriving the bilinear model using local expansion results such as Carleman linearization. This technique is accurate for capturing local nonlinearities around an operating point, but may be erroneous in describing global nonlinear behavior (Doyle III *et al.*, 1995). Since the discrete-time Volterra model was obtained from the continuous-time model, it also erroneously predicts a sign change in gain near $y = 18,000$ kg/kmol. The Volterra series is a time-invariant operator and would need parameter adaptation if it is to be used over a larger region of operation. Several researchers (Hernández and Arkun, 1993; Ricker and Lee, 1995) have incorporated on-line parameter adaptation into nonlinear MPC schemes, and this is a promising direction for future research.

4.2. Case Study II: MIMO control of a CSTR

The second case study considers the control of the free-radical solution polymerization of styrene in a jacketed CSTR. Hidalgo and Brosilow (1990) proposed a nonlinear MPC scheme to control the reactor temperature at the unstable steady state by manipulating the cooling water and monomer flow rates. For this example, two changes were made to the original problem. First, it is desired to control the number average molecular weight in addition to the reactor temperature to obtain a MIMO control

problem. To accomplish these objectives, the initiator and cooling water flow rates were selected as manipulated variables. A schematic of this process is shown in Fig. 9. The monomer flow rate was held constant. The second modification to the original problem is that the reactor is controlled around the low conversion stable steady-state point. Derivation of a bilinear model at a stable steady state enabled calculation of the model parameters using equations (38)–(41). The nonlinear model was obtained by augmenting the original four state model with two additional equations that are used to determine the number average molecular weight

$$\frac{d[I]}{dt} = \frac{(Q_i[I_f] - Q_t[I])}{V} - k_d[I], \quad (47)$$

$$\frac{d[M]}{dt} = \frac{(Q_m[M_f] - Q_t[M])}{V} - k_p[M][P], \quad (48)$$

$$\frac{dT}{dt} = \frac{Q_t(T_f - T)}{V} + \frac{(-\Delta H_r)}{\rho C_p} k_p[M][P] - \frac{hA}{\rho C_p V} (T - T_c), \quad (49)$$

$$\frac{dT_c}{dt} = \frac{Q_c(T_{cf} - T_c)}{V_c} + \frac{hA}{\rho_c C_{pc} V_c} (T - T_c), \quad (50)$$

$$\frac{dD_0}{dt} = 0.5k_t[P]^2 - \frac{Q_t D_0}{V}, \quad (51)$$

$$\frac{dD_1}{dt} = M_m k_p[M][P] - \frac{Q_t D_1}{V}, \quad (52)$$

$$y_1 = \frac{D_1}{D_0}, \quad (53)$$

$$y_2 = T. \quad (54)$$

where

$$[P] = \left[\frac{2fk_d[I]}{k_t} \right]^{0.5},$$

$$k_i = A_i \exp(-E_i/T), \quad i = d, p, t,$$

$$Q_t = Q_i + Q_s + Q_m.$$

Two assumptions were made in deriving equations (51) and (52). First, it was assumed that the rate of disappearance of monomer was primarily due to propagation. Hence, disappearance of monomer due to chain transfer to monomer was not included in the model. This assumption was also made in Hidalgo and Brosilow (1990) and Jaisinghani and Ray (1977). The second assumption was that the overall chain termination rate constant, k_t , was composed of both combination, k_{Tc} , and disproportionation, k_{Td} , contributions (Schmidt and Ray, 1981) where

$$k_t = k_{Tc} + k_{Td}. \quad (55)$$

For styrene, chain termination has been experimentally determined to occur solely by combination in bulk (Bevington *et al.*, 1954) and solution (Timm

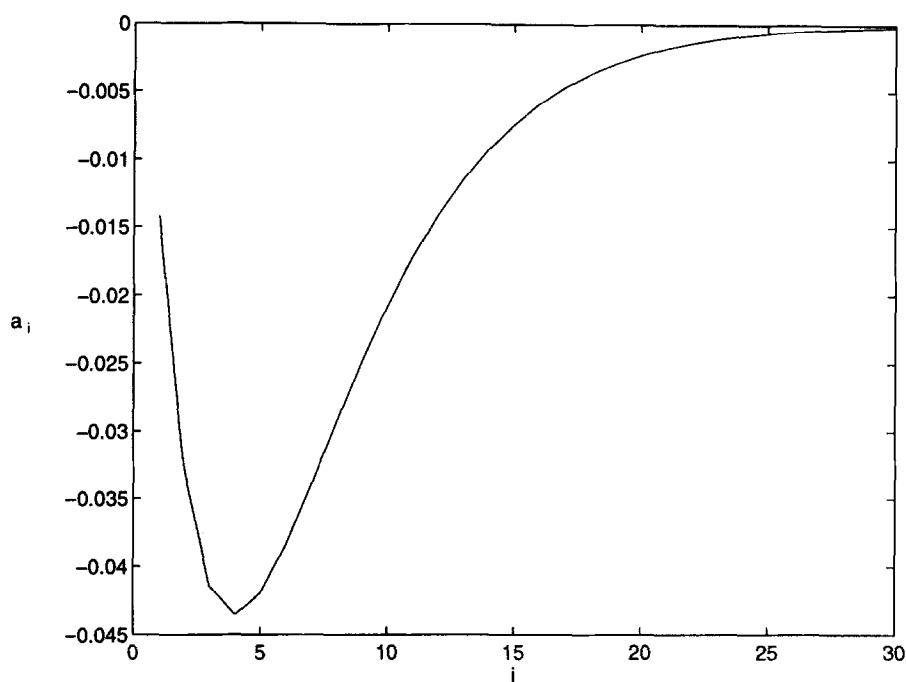


Fig. 5. First-order coefficients for Case Study I.

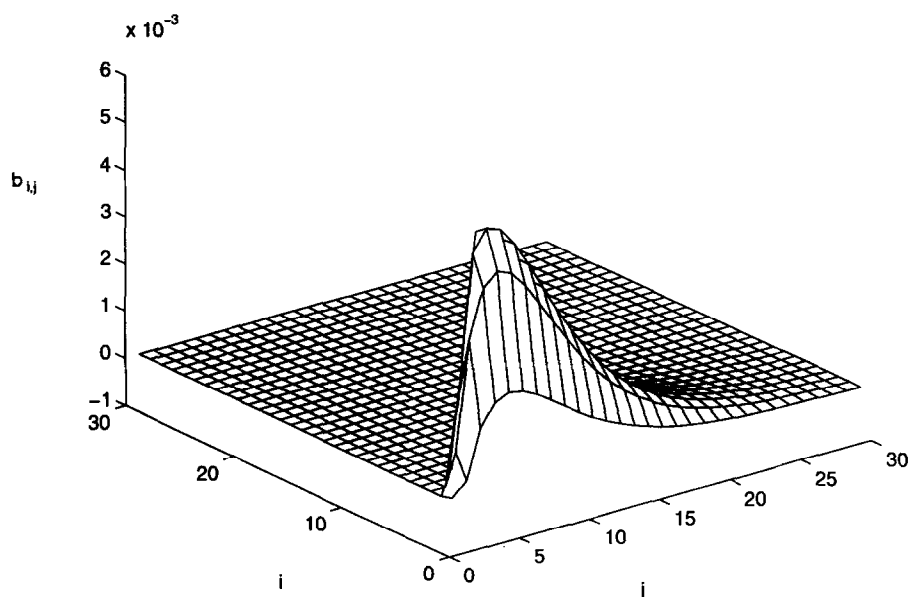


Fig. 6. Second-order coefficients for Case Study I.

and Rachow, 1974) polymerization. Hence, $k_t = k_{T_c}$ in equation (55). The kinetic and thermodynamic parameters (which correspond to Process 1 in the original paper (Hidalgo and Brosilow, 1990)), model parameters, and steady-state operating conditions are listed in Tables 4–6, respectively. The model assumptions affecting the parameter selection are the same as those in the original paper (Hidalgo and Brosilow, 1990). The first assumption is that the solvent volume fraction is main-

tained at 0.6 so that the gel effect may be neglected (Choi, 1986). Hence, the solvent flow rate was manipulated according to the following equation:

$$Q_s = 1.5Q_m - Q_i. \quad (56)$$

A second model assumption is that the temperature must be less than 423 K. This is justified by the fact that the rate of thermal initiation becomes significant at temperatures greater than 373 K, and prob-

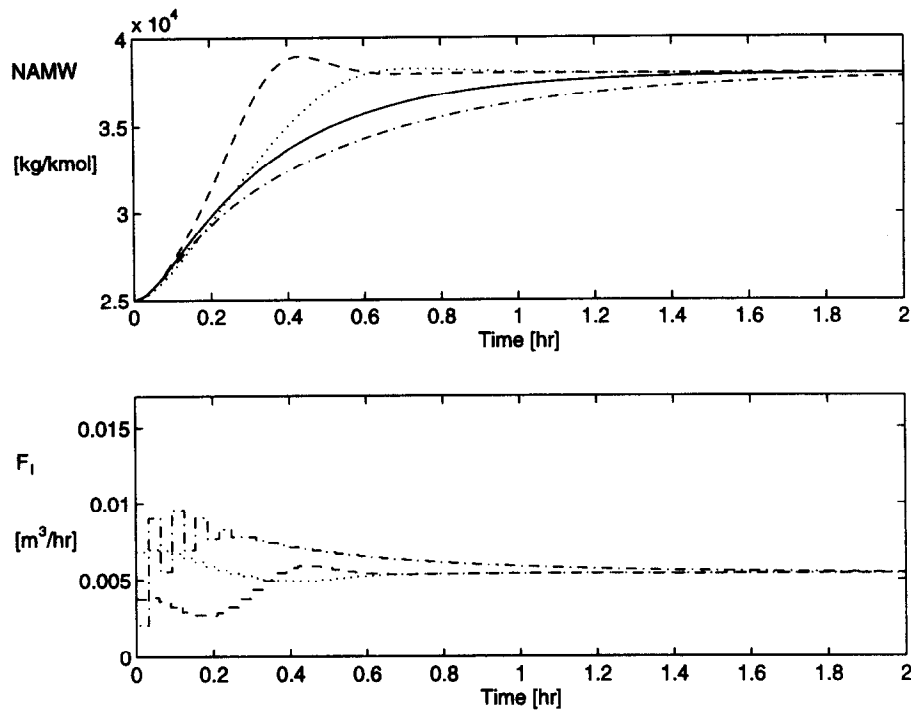


Fig. 7. Closed-loop simulation for a filtered step setpoint change from 25,000.5 to 38,000 kg/kmol using the reduced model. Solid: reference; dashed: linear MPC; dotted: Volterra MPC; dash dot: nonlinear QDMC.

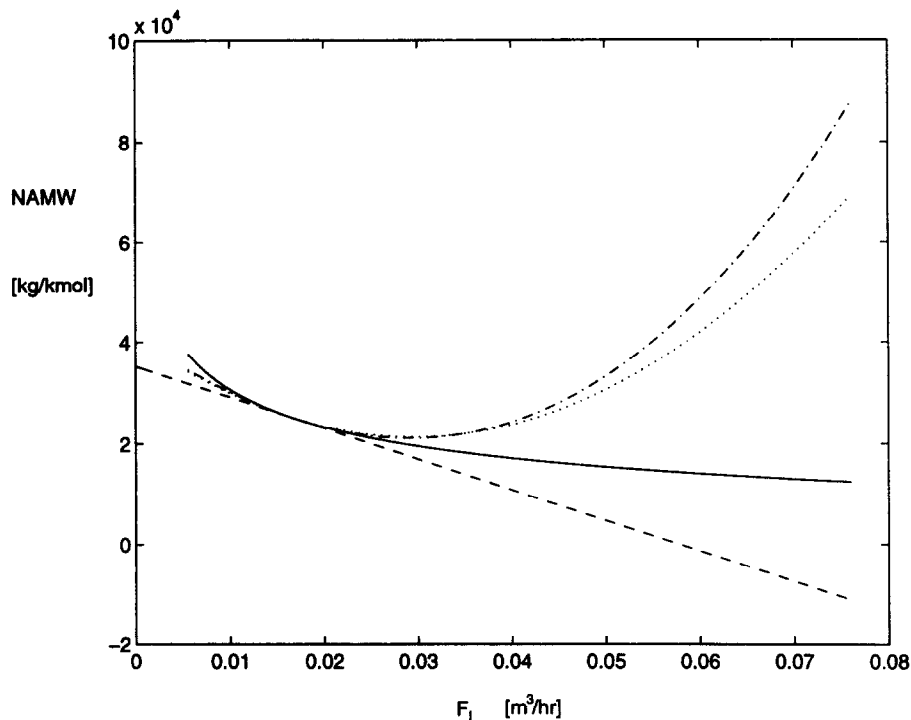


Fig. 8. Gain loci for various models. Solid: nonlinear; dashed: linear; dotted: Volterra; dash dot: bilinear.

ably dominates catalytic initiation at temperatures above 423 K (Biesenberger and Sebastian, 1983).

The goal of the control system is to drive the polymerization system to a new state to produce polymers with different number average molecular

weights while keeping the temperature at its setpoint. The vectors u , x , and y were placed in deviation form, and a bilinear representation of the nonlinear model was again obtained using Carleman linearization (Rugh, 1981). The model matri-

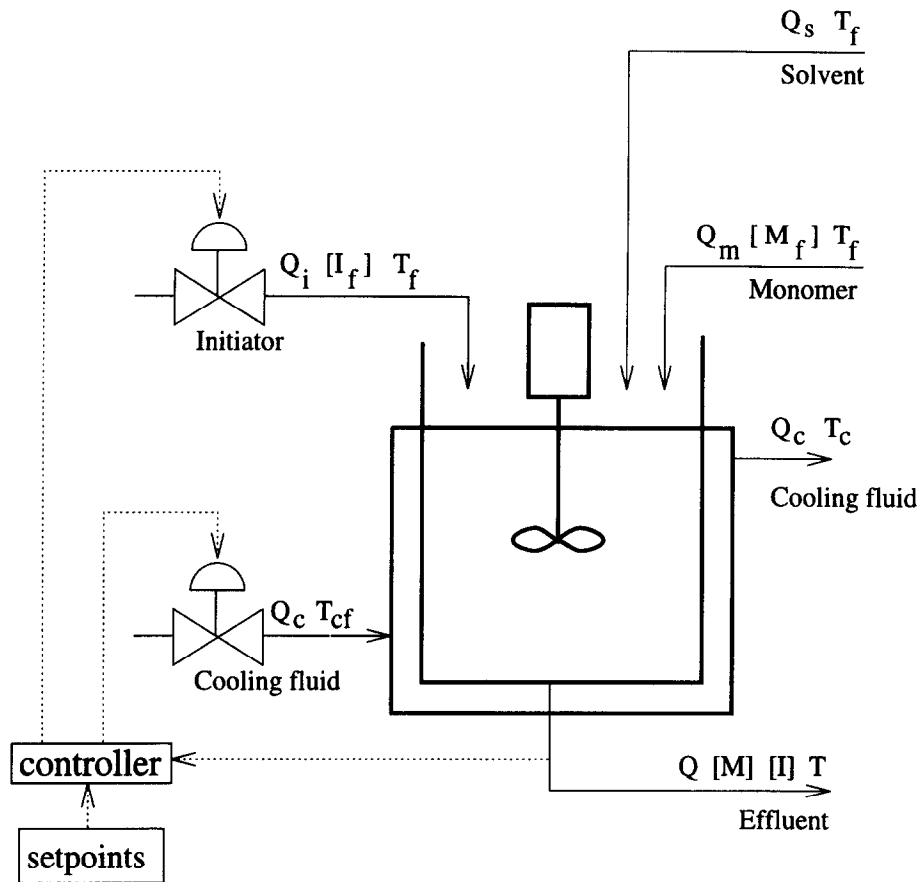


Fig. 9. Control configuration for Case Study II.

Table 4. Kinetic and thermodynamic parameters for Case Study II

f	=	0.6	
A_d	=	5.95×10^{13}	1/s
E_d	=	14,897	K
A_t	=	1.25×10^9	$\ell/(\text{mol s})$
E_t	=	843	K
A_p	=	1.06×10^7	$\ell/(\text{mol s})$
E_p	=	3557	K
$-\Delta H_r$	=	16,700	cal/mol
hA	=	70	cal/(K s)
ρC_p	=	360	cal/(K ℓ)
ρC_{pc}	=	966.3	cal/(K ℓ)

Table 5. Parameters for Case Study II

Q_s	=	0.1275	ℓ/s
Q_m	=	0.105	ℓ/s
V	=	3000	ℓ
V_c	=	3312.4	ℓ
$[I_f]$	=	0.5888	mol/ ℓ
$[M_f]$	=	8.6981	mol/ ℓ
T_f	=	330	K
T_{cf}	=	295	K
M_m	=	104.14	g/mol

ces were scaled with the following scaling matrices:

$$\mathbf{N}_u = \begin{bmatrix} 108 & 0 \\ 0 & 471.6 \end{bmatrix}, \quad (57)$$

$$\mathbf{N}_y = \begin{bmatrix} 2500 & 0 \\ 0 & 0.5 \end{bmatrix}. \quad (58)$$

The scaled, continuous-time bilinear system was discretized using the explicit fourth-order Runge–Kutta method given by equations (32)–(35) with a sampling time of 1 h. Although the sampling interval is large, the sampling time is still less than one-fifth the smallest time constant in the linear transfer function matrix. The open-loop and closed-loop responses are also of the same order as those observed in the control of a 1000 ℓ polymerization reactor in Congalidis *et al.* (1989). The largest negative eigenvalue of the Hidalgo and Brosilow system was -0.74 h^{-1} at the low conversion steady-state. The large time constant for this system arises from the large reactor volume of 3000 ℓ and operation of the reactor at the low temperature, low conversion steady state. The model parameters were calculated using equations (38)–(41). The Volterra controller described in Section 2.2 was used in this case study, i.e. a fourth-order nonlinear program was solved at each sampling interval. The tuning parameters for linear and Volterra MPC were $M = 1$, $P = 20$, $[\gamma_1, \gamma_2] = [2, 1]$, and $[\lambda_1, \lambda_2] = [1000, 1]$. The truncation order of the model was $N = 35$. The truncation order was chosen to obtain a model with a mem-

ory approximately equal to three times the largest time constant of this process. The plant was the actual nonlinear model integrated with the Gear predictor-corrector method.

Table 6. Steady-state operating conditions for Case Study II

$x_1 = [I]$	=	6.6832×10^{-2}	mol/l
$x_2 = [M]$	=	3.3245	mol/l
$x_3 = T$	=	323.56	K
$x_4 = T_c$	=	305.17	K
$x_5 = D_0$	=	2.7547×10^{-4}	mol/l
$x_6 = D_1$	=	16.110	g/l
$u_1 = Q_i$	=	0.03	l/s
$u_2 = Q_c$	=	0.131	l/s
y_1	=	58,481	g/mol
y_2	=	323.56	K

Open-loop responses of the nonlinear, linear, and second-order Volterra models to step changes of ± 27 l/h in Q_i from its nominal value of 108 l/h are shown in Fig. 10. In both simulations, the output of the second-order Volterra model more closely tracks the actual nonlinear plant output for the number average molecular weight (and the curves are indistinguishable for one of the step changes). Although these step changes correspond to $\pm 25\%$ changes in the initiator flow rate, the temperature does not significantly deviate from the nominal value. Both the linear and second-order Volterra models accurately predict the moderate temperature changes of the plant.

The result of a closed-loop simulation for a setpoint change from 58,481 to 80,000 g/mol is shown in Fig. 11. As in the preceding example, the Volterra model predictive controller outperforms the linear model predictive controller, as the number average molecular weight more closely tracks the reference trajectory. Conversely, linear MPC results in significant overshoot. The overshoot is caused by the more aggressive manipulated variable profiles in Fig. 11. Linear MPC may be detuned to yield an overdamped response for the number average molecular weight. However, improved performance for positive setpoint changes would be achieved at the expense of performance deterioration for negative setpoint changes.

The dash dot lines in Fig. 11 correspond to input and output profiles obtained using nonlinear QDMC with $M = 1$, $P = 12$, $[y_1, y_2] = [1, 2000]$, and $[\lambda_1, \lambda_2] = [0, 0]$. Different tuning parameters were used for nonlinear QDMC as compared to those used for linear and Volterra MPC, because the underlying linear models were different from those used in the other two controllers. The tuning parameters used in nonlinear QDMC resulted in an improvement in closed-loop performance as compared to incorporating the tuning parameters used in linear and Volterra MPC.

The performance obtained using nonlinear QDMC is comparable to that of using Volterra MPC. The response of the number average molecular weight is approximately first order, and the temperature is maintained close to its setpoint. In addition, the input profiles for both flow rates are very smooth. The reason for the good performance of nonlinear QDMC in this case study is that the first move in initiator flow rate is -18.19% from its nominal value, and the first move in cooling water flow rate is -18.55% from its nominal value. Hence, the input variable changes are much smaller, on a percentage basis, than those encountered in Case Study I. The linear models obtained at each sampling interval are more accurate when small input variable changes are used, and the corresponding control action is more appropriate. An attractive feature of nonlinear QDMC is that a single quadratic program needs to be solved on-line, while the second-order Volterra controller requires the solution of a fourth-order nonlinear program. However, the nonlinear QDMC algorithm requires the availability of a fundamental model. In addition, the nonlinear QDMC results presented in both case studies assume there is no plant-model mismatch, which will not be the case in practice.

The simulation in Fig. 11 used a triangular second-order Volterra model that did not include cross terms ($u_1(k-1)u_2(k-1)$, ..., etc.) in the inputs. For a truncation order of $N = 35$, this resulted in 2520 second-order coefficients. The first and second-order parameters for this case study are shown in Figs 12 and 13, respectively. The first-order parameters are approximately one order of magnitude larger than the second-order terms, and the scaled manipulated variables were on the order of 10^{-1} . Hence, the later $b_{i,j,l,n}^i$ terms in Fig. 13 make a very small contribution to the model. Although the reactor temperature is affected by the heat of polymerization and the Arrhenius temperature dependence of the kinetic parameters, the large reactor acts as a heat sink and prevents a large change in temperature provided there is always a nominal amount of cooling water flowing through the cooling jacket. Hence, linear models could be used to describe the relationship between the two inputs and the reactor temperature. In addition, the initiator flow rate has a much greater effect on the number average molecular weight than the cooling water flow rate, and a linear model could also be used to relate the cooling water flow rate and the number average molecular weight. In addition, if the truncation order of the second-order parameters in the remaining Volterra model is reduced from $N = 35$ to 20, using the same reasoning in the first case study, the number of second-order coefficients is reduced to 210. Repeating the same

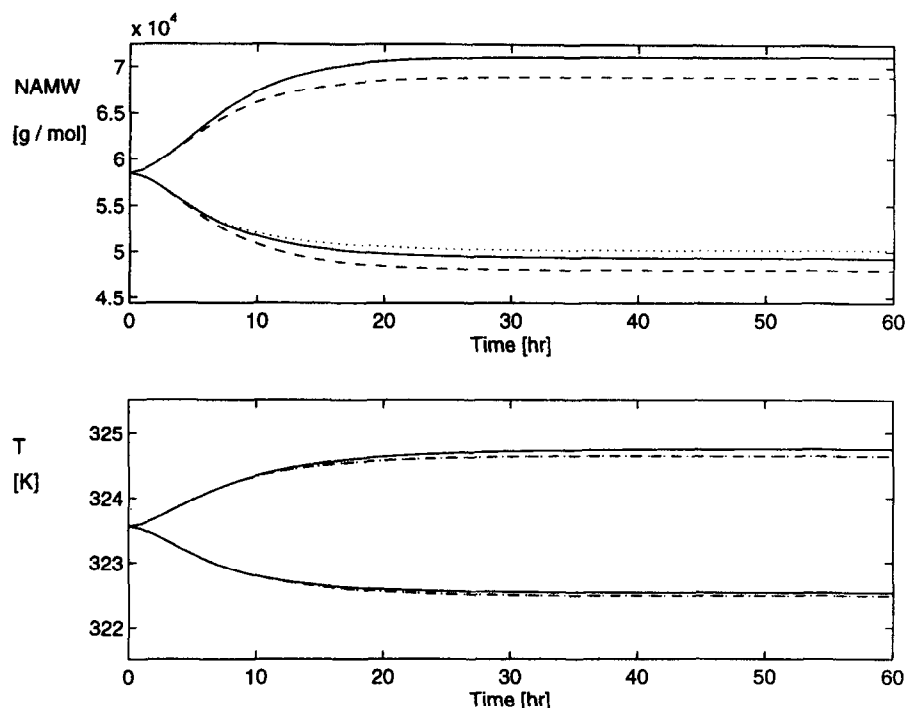


Fig. 10. Open-loop simulations for step changes of $\pm 27 \text{ l/h}$ in Q_j from its nominal value of 108 l/h . Solid: nonlinear; dashed: linear; dotted: Volterra.

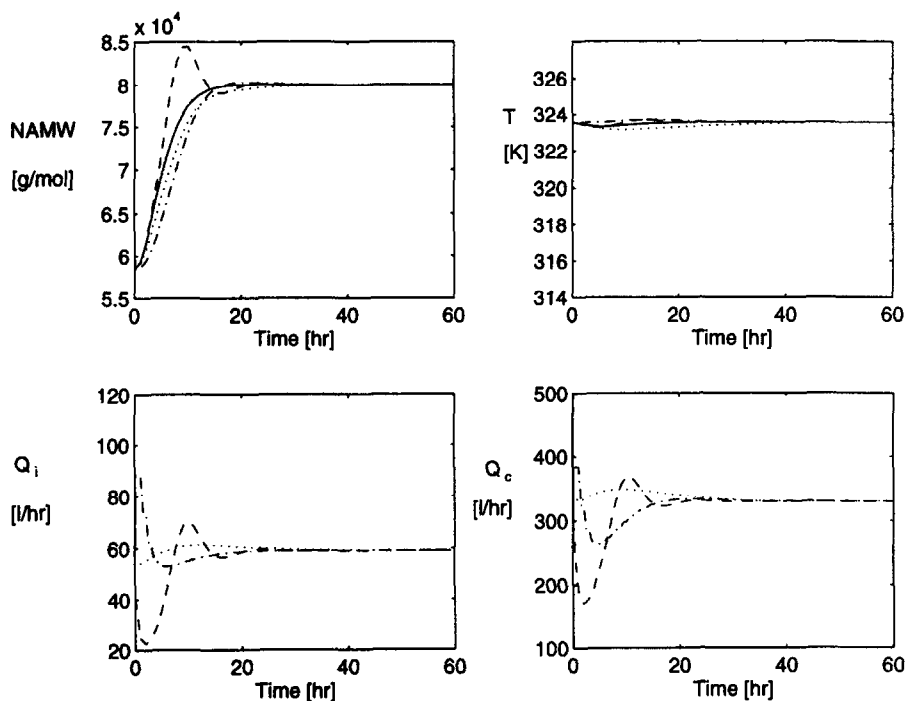


Fig. 11. Closed-loop simulation for a step setpoint change from 58,481 to 80,000 g/mol. Solid: reference; dashed: linear MPC; dotted: Volterra MPC; dash dot: nonlinear QDMC.

setpoint change simulation yielded the closed-loop performance illustrated in Fig. 14. The closed-loop performance does not significantly degrade with the elimination of over 90% of the second-order parameters in the second-order Volterra model.

One of the drawbacks of this approach is that

second-order Volterra models require many parameters to describe nonlinearities. In both case studies, it was shown that the number of second-order coefficients could be significantly reduced with no significant difference in closed-loop performance. While it seems reasonable that the memory

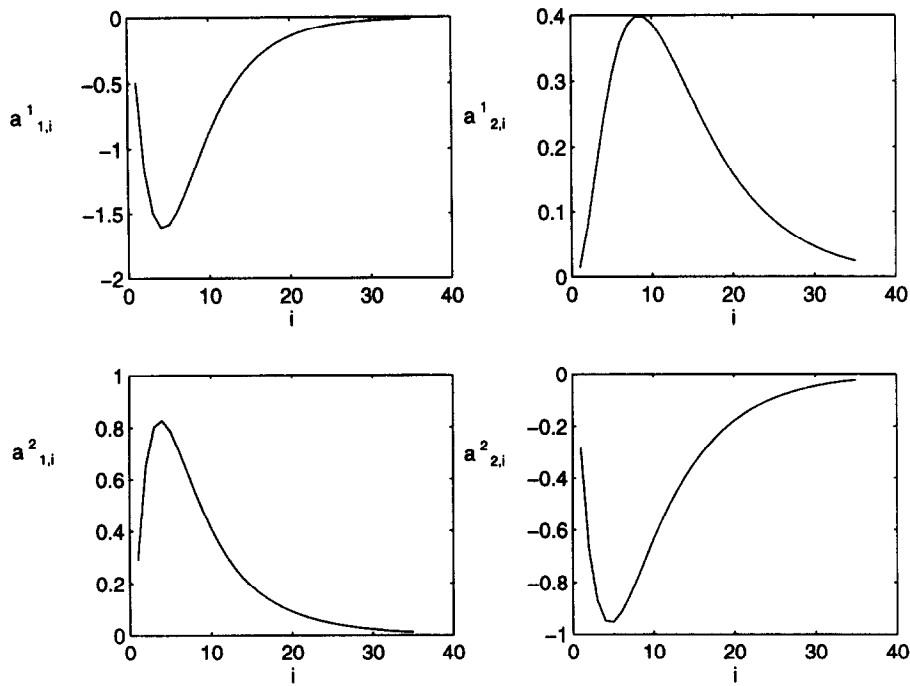


Fig. 12. First-order coefficients for Case Study II.

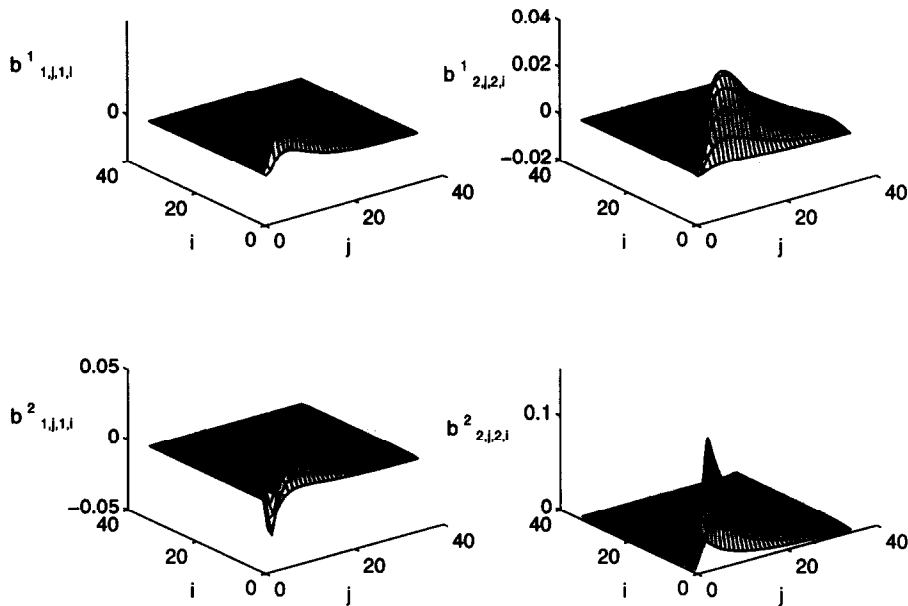


Fig. 13. Second-order coefficients for Case Study II.

of the linear and second-order contributions to the system dynamics should be approximately the same, the $b^i_{l,j,l,n}$ parameters are one order of magnitude smaller than the $a^i_{l,j}$ coefficients, and act to correct the linear model for the effect of nonlinearities. Hence, the later $b^i_{l,j,l,n}$ terms are very small and may be omitted without significant performance degradation in modeling and control. Another disadvantage is associated with using a

second-order nonlinear model. The performance of a linear model-based controller usually degrades as the system is moved away from the point at which the model was obtained. Although performance degradation occurs, the linear model-based controller is still stable over a fairly broad regime. The performance of a second-order nonlinear model-based controller also typically worsens as the plant is moved away from the point at which the model

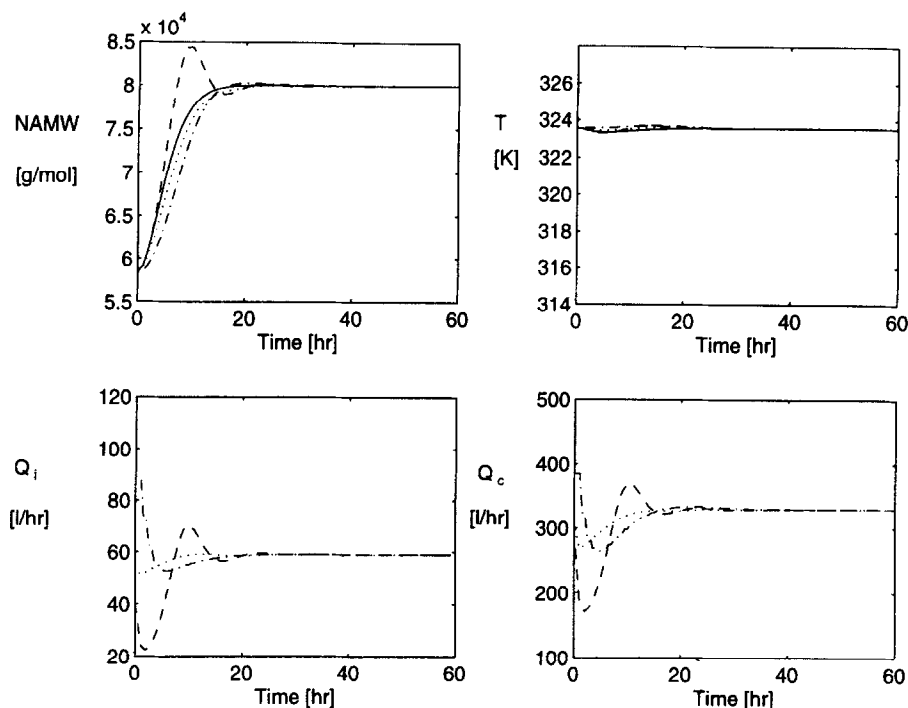


Fig. 14. Closed-loop simulation for a step setpoint change from 58,481 to 80,000 g/mol using the reduced model. Solid: reference; dashed: linear MPC; dotted: Volterra MPC; dash dot: nonlinear QDMC.

was obtained. However, due to the inherent curvature of second-order models, they will always predict that the plant gain will change sign at some point. Hence, care should be taken to ensure that a second-order model-based controller is not used outside the region for which the model is valid. The ability of a second-order Volterra model derived from Carleman linearization to capture 'local' and not global behavior was also demonstrated in Doyle III *et al.* (1995).

Despite the drawbacks associated with the proposed second-order Volterra controller, there are also several advantages of this control scheme. One advantage is that improved performance over a linear model-based controller can be achieved as illustrated in both case studies. A second advantage is that these models yield a fourth-order nonlinear program for the standard two-norm MPC objective function if the controller is implemented as shown in Section 2.2. While this optimization problem is more difficult to solve than the quadratic program encountered using traditional linear MPC, it is less computationally burdensome than the nonlinear program that arises in modified nonlinear internal model control or nonlinear MPC using polynomial ARMA models. A third advantage results from the fact that Volterra models are based only on past inputs. Hence, stability results are easier to obtain than for models based on past inputs and outputs. Stability results for the continuous-time inverse Volterra series (Zheng and Zafriou, 1993) and the one-norm SISO MPC formulation (Genceli

and Nikolaou, 1995) have been presented by other researchers. In addition, the ability to obtain the second-order Volterra model parameters from Carleman linearization and input-output data is another advantage.

5. CONCLUSIONS

The ability of second-order Volterra models to capture asymmetric changes in output given symmetric changes in input was illustrated with open-loop simulations. A nonlinear MPC scheme utilizing a successive substitution algorithm was presented. This controller yielded improved performance over linear MPC for the control of number average molecular weight in an isothermal CSTR. Improved multivariable control of the number average molecular weight and temperature in a large polymerization reactor was also demonstrated. The models were obtained by Carleman linearization of the fundamental model, discretization of the resulting continuous-time bilinear model, and calculation of the parameters for a triangular second-order Volterra model. Relationships between discrete-time bilinear model matrices and their continuous-time counterparts using an explicit fourth-order Runge-Kutta method were also presented. It was also demonstrated that a large reduction in the number of second-order parameters could be made without a significant degradation in closed-loop performance.

Acknowledgements—FJD and BRM would like to acknowledge funding from the National Science Foundation NYI program (Grant CTS-9257059). Additional support for BRM was provided by a Purdue University Andrews Fellowship.

REFERENCES

- Bevington, J. C., H. W. Melville and R. P. Taylor (1954). The termination reaction in radical polymerizations. II Polymerizations of styrene at 60° and of methyl methacrylate at 0 and 60°, and the copolymerization of these monomers at 60°. *J. Polym. Sci.*, **XIV**, 463–476.
- Biesenberger, J. A. and D. H. Sebastian (1983). *Principles of Polymerization Engineering*. John Wiley & Sons, New York.
- Carnahan, B., H. A. Luther and J. O. Wilkes (1969). *Applied Numerical Methods*. John Wiley & Sons, New York.
- Choi, K. Y. (1986). Analysis of steady state of free radical solution polymerization in a continuous stirred tank reactor. *Polymer Engng. Sci.*, **26**(14), 975–981.
- Chow, T., E. Eskow and R. Schnabel (1994). Algorithm 739: A software package for unconstrained optimization using tensor methods. *ACM Trans. Math. Software*, **20**(4), 518–530.
- Congalidis, J. P., J. R. Richards and W. H. Ray (1989). Feedforward and feedback control of a solution copolymerization reactor. *AIChE J.*, **35**(6), 891–907.
- Daoutidis, P., M. Soroush and C. Kravaris (1990). Feedforward/feedback control of multivariable nonlinear processes. *AIChE J.*, **36**(10), 1471–1484.
- Doyle III, F. J., B. A. Ogunnaike and R. K. Pearson (1995). Nonlinear model-based control using second-order Volterra models. *Automatica*, **31**(5), 697–714.
- Economou, C. G. (1985). An operator theory approach to nonlinear controller design. PhD thesis, Chemical Engineering, California Institute of Technology, Pasadena, CA.
- Economou, C. G. and M. Morari (1985). Newton control laws for nonlinear controller design. In *Proc. IEEE Conf. on Decision and Control*, Fort Lauderdale, FL, pp. 1361–1366.
- García, C. E. (1984). Quadratic dynamic matrix control of nonlinear processes. An application to a batch reaction process. *AIChE Annual Meeting*, San Francisco, CA.
- Gattu, G. and E. Zafiriou (1992). Nonlinear quadratic dynamic matrix control with state estimation. *Ind. Engng Chem. Res.*, **31**, 1096–1104.
- Genceli, H. and M. Nikolaou (1995). Design of robust constrained model-predictive controllers with Volterra series. *AIChE J.*, **41**(9), 2098–2107.
- Hernández, E. (1992). Control of nonlinear systems using input-output information. PhD thesis, Chemical Engineering, Georgia Institute of Technology, Atlanta, GA.
- Hernández, E. and Y. Arkun (1993). Control of nonlinear systems using polynomial ARMA models. *AIChE J.*, **39**(3), 446–460.
- Hidalgo, P. M. and C. B. Brosilow (1990). Nonlinear model predictive control of styrene polymerization at unstable operating points. *Computers Chem. Engng*, **14**(4/5), 481–494.
- Jaisinghani, R. and W. H. Ray (1977). On the dynamic behaviour of a class of homogeneous continuous stirred tank polymerization reactors. *Chem. Engng Sci.*, **32**, 811–825.
- Lee, J. H. and N. L. Ricker (1993). Extended Kalman filter based nonlinear model predictive control. In *Proc. Am. Control Conf.*, San Francisco, CA, pp. 1895–1899.
- Li, W. C. and L. T. Biegler (1988). Process control strategies for constrained nonlinear systems. *Ind. Engng Chem. Res.*, **27**, 1421–1433.
- Li, W. C. and L. T. Biegler (1989). Multistep, Newton-type control strategies for constrained, nonlinear processes. *Chem. Engng Res. Des.*, **67**, 562–577.
- Maner, B. R. (1993). Nonlinear model predictive control with second order Volterra models. Master's thesis, Chemical Engineering Dept, Purdue University, West Lafayette, IN.
- Pearson, R. K., B. A. Ogunnaike and F. J. Doyle III (1996). Identification of discrete convolution models for nonlinear processes. *IEEE Trans. Acoustics, Speech, and Signal Processing*, in press.
- Pearson, R. K., B. A. Ogunnaike and F. J. Doyle III (n.d.). Second-order Volterra models for chemical processes, Part I. In preparation.
- Ricker, N. L. (1985). Use of quadratic programming for constrained internal model control. *Ind. Engng Chem. Process Des. Dev.*, **24**, 925–936.
- Ricker, N. L. and J. H. Lee (1995). Nonlinear model predictive control of the Tennessee Eastman challenge process. *Computers Chem. Engng*, **19**(9), 961–981.
- Rugh, W. J. (1981). *Nonlinear System Theory — The Volterra/Wiener Approach*. The Johns Hopkins University Press, Baltimore, MD.
- Schmidt, A. D. and W. H. Ray (1981). The dynamic behavior of continuous polymerization reactors—I Isothermal solution polymerization in a CSTR. *Chem. Engng Sci.*, **36**, 1401–1410.
- Timm, D. C. and J. W. Rachow (1974). Description of polymerization dynamics by using population density. In H. M. Hulburt (Ed.), *Chemical Reaction Engineering—II*, Advances in Chemistry Series 133. American Chemical Society, pp. 122–136.
- Zames, G. (1966). On the input-output stability of time-varying nonlinear feedback systems, part I: Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE Trans. Automat. Control*, **AC-11**(2), 228–238.
- Zheng, Q. and E. Zafiriou (1993). Stability analysis of inverse Volterra series. Presented at the *AIChE Annual Meeting*, St. Louis, MO.