



Nonlinear Model-based Control Using Second-order Volterra Models*

FRANCIS J. DOYLE III,^{†‡} BABATUNDE A. OGUNNAIKE[†] and RONALD K. PEARSON[†]

Analysis, synthesis and design procedures are presented for nonlinear model-based control employing a Volterra model—a nonlinear extension of the linear convolution model used in conventional model predictive control.

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

Abstract—A nonlinear controller synthesis scheme is presented that retains the original spirit and characteristics of conventional (linear) model predictive control (MPC) while extending its capabilities to nonlinear systems. The scheme employs a Volterra model—a simple and convenient nonlinear extension of the linear convolution model employed by conventional MPC—and gives rise to a controller composed of a conventional linear controller augmented by an auxiliary loop of nonlinear ‘corrections’. Simulation case studies involving two different examples representative of the typical spectrum of nonlinear behavior in real chemical processes—an industrial polymerization reactor and an isothermal reactor exhibiting inverse response—are used to demonstrate the practical utility of the control scheme and to evaluate its performance.

1. INTRODUCTION

Linear controller design techniques are widely employed in the chemical/petrochemical industries. In particular, conventional model predictive control (MPC) has enjoyed widespread acceptance and success as an effective technique for dealing with difficult control problems of industrial importance. The appeal of MPC (and other linear techniques) is partly due to the simplicity arising from the linear models used to approximate process behavior. However, this also constitutes a potential deficiency, because such simple linear models are often inadequate if a more realistic approximation of the usually complex, nonlinear processes is desired. On the other hand, the nonlinear control schemes that employ more realistic—and often more complex—nonlinear process descriptions typically sacrifice the simplicity associated

with linear techniques in order to achieve improved performance (cf. Eaton and Rawlings, 1990; Zafriou and Marchal, 1991; Beigler and Rawlings, 1991). More importantly, these nonlinear techniques require nonlinear differential equation process models. This limits industrial application, since such first principles models are not readily available in industrial practice (because of a chronic lack of detailed and extensive process knowledge required for their development).

In this paper we present a direct synthesis scheme for nonlinear model-based control that employs a Volterra model: a nonlinear process representation obtainable from input/output data and similar in form to the conventional MPC model. Furthermore, the scheme gives rise to a controller that clearly decomposes into a conventional linear model-based controller plus a sequence of analytical nonlinear ‘perturbation’ or correction terms. In particular, when the process representation is a second-order Volterra model, the control structure is decomposed into two elements:

- (i) an optimal linear controller;
- (ii) a loop with a second-order Volterra term to account for the process nonlinearity.

The first step involves the typical linear model inversion that characterizes conventional model-based control. The notion of ‘inversion’ that is used here is fairly general and can be relaxed to incorporate penalties on manipulated variables and receding horizon performance criteria (see e.g. Richalet *et al.*, 1978; García, 1984). The control engineer is at liberty to select the linear strategy for implementing this step; the correction obtained in the second step involves only the second-order Volterra plant model. Thus, only the linear inversion involved in obtaining the linear controller is required to synthesize the compensator for the full second-

* Received 15 September 1993; received in final form 2 August 1994. This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor J. C. Kantor under the direction of Editor Yaman Arkun. Corresponding author Dr Babatunde A. Ogunnaike. Tel. +1 302 695 2535; Fax +1 302 695 2645; E-mail ogunnaike@essptd.dnet.dupont.com.

[†] E. I. DuPont deNemours & Co., Inc., Experimental Station El, Wilmington, DE 19880-0101, U.S.A.

[‡] Present address: School of Chemical Engineering, Purdue University, West Lafayette, IN 47907-1283, U.S.A.

order Volterra model. For higher-order Volterra models, the successively higher-order correction terms (third, fourth etc.) required in addition to the linear controller are implemented in a similarly straightforward manner.

The idea of employing second-order and bilinear models for model-based control has been explored by a number of investigators. Yeo and co-workers (Yeo and Williams, 1987; Lo *et al.*, 1992) have formulated a nonlinear MPC algorithm for bilinear process model descriptions. No formal approach for the identification of their model is presented, and the solution of the resulting optimization problem still requires nonlinear programming techniques. Recently, Bartee and Georgakis (1992) have investigated Volterra modeling for process control employing a reference system synthesis technique for controller design, and an identification procedure that utilizes PRBS signals to generate a bilinear model from which the equivalent Volterra model is then generated. The idea of using Volterra series to calculate analytically the control law required to eliminate successively higher-order effects in the closed-loop is originally due to Sain and Al-Baiyat (Al-Baiyat, 1986; Al-Baiyat and Sain, 1986). Explicit expressions for the time-domain realizations of the Laplace-domain controller are determined from bilinear approximation of a first-principles model, and explicit expressions for its time-domain realization are presented; the limitations imposed by non-invertible plants are not addressed, however.

The nonlinear model-based scheme we propose essentially relies on an arbitrary linear model-based controller for a foundation and augments the linear control solution with analytical, second- and higher-order correction terms, which are implemented in the form of an auxiliary loop around the linear controller. Such a scheme has the advantage of providing an extra level in the hierarchical structure available to the control room operator: instead of having 'manual' and 'automatic' as the only options, the operator now has the additional option of switching off only the auxiliary nonlinear loop, and downgrade, if necessary, not all the way to manual, but first to the basic linear scheme. In contrast to the extended DMC approach (Simminger *et al.*, 1991, Peterson *et al.*, 1992), which also employs a perturbation approach to nonlinear model predictive control, the approach we present here requires no iterations for obtaining the correction term.

The fact that only a linear inversion is required in the synthesis of this controller is a most attractive characteristic of the scheme; it also underscores the well-known performance

limitation imposed on all control strategies (linear and nonlinear), namely the invertibility of the underlying process model. Thus the issue of how to handle, within this framework, nonlinear systems that suffer from invertibility problems is also discussed. Not surprisingly, the resulting controller has the exact model inverse replaced by an appropriate pseudo-inverse. We address in a different paper the important and closely related issue of how the Volterra models upon which this scheme is based may be identified from plant input/output data (cf. Pearson *et al.*, 1992).

The paper is organized as follows. In Section 2 a partition model structure—of which the Volterra model is a specific example—is introduced as the basis of the direct synthesis scheme for nonlinear control. In Section 3 a brief review of Volterra series is presented along with strategies for obtaining Volterra series models. In Section 4 a control synthesis strategy that utilizes a Volterra series model is discussed in detail. In Section 5 this control approach is applied to two different chemical reactors that are representative of the range of practical nonlinear process control problems. Conclusions and a summary follow in Section 6.

2. NONLINEAR CONTROLLER SYNTHESIS

2.1. General concepts

Consider a process whose dynamic behavior is represented by the nonlinear input/output correlation implied by the model:

$$\hat{y} = \mathbf{P}[u], \quad (1)$$

where \mathbf{P} is a general nonlinear operator that maps the input u into the output, or response, \hat{y} . If y represents the actual measurement of the plant output then the model error obtained as

$$e = y - \hat{y} \quad (2)$$

enables us to write

$$y = \mathbf{P}[u] + e \quad (3)$$

as the relationship between the plant input and the actual plant output.

Given y^* as the desired trajectory for the actual plant output y to follow, the control action u , that satisfies the objective

$$\min_u \phi = \|y^* - y\| \quad (4)$$

is easily obtained as

$$u = \mathbf{P}^{-1}[y^* - e]$$

provided the inverse of the operator \mathbf{P} exists. If y^* is chosen as y_d , the set-point for y , then

$$u = \mathbf{P}^{-1}[y_d - e]. \quad (5)$$

Note here that nominal stability is concerned with the case $e=0$, for which (5) implies an open-loop control policy; feedback appears only in the presence of modeling error and unmeasured disturbances ($e \neq 0$).

There are several theoretical issues raised by such a synthesis technique, which are better confronted later when we consider a specific form for the nonlinear operator \mathbf{P} ; in the meantime we note that, in general,

- (i) \mathbf{P}^{-1} may not exist; and when it does, it may not be realizable;
- (ii) even when \mathbf{P}^{-1} exists (and is realizable), it is clear that (5) results in so-called 'perfect control', with all the attendant robustness and implementation problems;
- (iii) the model error e is composed of an essentially inseparable combination of unmodelled dynamics, unmeasured disturbances and noise components; responding to this signal directly as implied in (5) will result in serious robustness problems.

The standard approach to this problem is the introduction of a set-point 'filter' defined by

$$y^* = \mathbf{F}[y_d]$$

and a systematic 'error estimator' \mathbf{E} , which produces \tilde{e} (given y , \mathbf{P} and u), and a resulting \tilde{y} , according to

$$\tilde{e} = \mathbf{E}[y, \mathbf{P}, u],$$

$$\tilde{y} = \mathbf{P}[u] + \tilde{e}.$$

Then, by modifying the objective in (4) using \tilde{y} in place of y , the result in (5) is modified to read

$$u = \mathbf{P}^{-1}[\mathbf{F}[y_d] - \tilde{e}]. \quad (6)$$

In principle, the two operators \mathbf{E} and \mathbf{F} may be chosen appropriately to address the problem of realizability, robustness and implementation raised by the requirement of the operator inverse in (5).

Thus the controller synthesis problem as posed here involves, conceptually, the construction of \mathbf{P}^{-1} given \mathbf{P} , and the choice of \mathbf{E} and \mathbf{F} . The resulting control system is represented in block diagrammatic form in Fig. 1. The choice of \mathbf{E} and \mathbf{F} and their implementation clearly depend

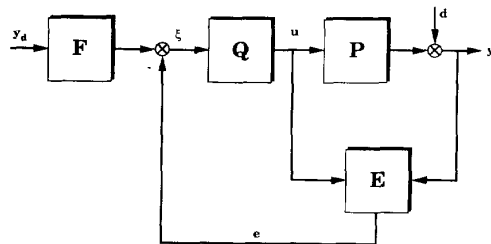


Fig. 1. General model-based control structure.

on the nature of \mathbf{P} and its inverse; however, no useful general statement can be made either about conditions for the existence of the inverse of the general nonlinear operator \mathbf{P} or about the procedure for its construction and implementation, without considering some special cases. Nevertheless, we note that if \mathbf{P} is a purely linear operator \mathbf{L} , then fairly general statements can be made about the nature, construction and implementation of its inverse; furthermore, if \mathbf{E} is chosen simply as a linear filter operating on the error e , and in addition this filter is chosen to be the same as the set-point filter, i.e.

$$\tilde{e} = \mathbf{F}[e],$$

and if this \mathbf{F} is chosen to have as one of its properties

$$\lim_{t \rightarrow \infty} \mathbf{F}[x(t)] = \lim_{t \rightarrow \infty} x(t), \quad (7)$$

then (6) becomes

$$u = \mathbf{P}^{-1}\mathbf{F}[y_d - e] \quad (8)$$

which is immediately recognizable as the standard IMC strategy (cf. Frank, 1974; García and Morari, 1982).

2.2. The partitioned nonlinear model

Consider the case in which the nonlinear operator \mathbf{P} can be partitioned into a linear and nonlinear portion as follows:

$$\mathbf{P} = \mathbf{L} + \mathbf{N}. \quad (9)$$

Then if the inverse of the linear portion exists, a simple rearrangement yields

$$\mathbf{P} = \mathbf{L}(\mathbf{I} + \mathbf{L}^{-1}\mathbf{N}), \quad (10)$$

from which we obtain

$$\mathbf{P}^{-1} = (\mathbf{I} + \mathbf{L}^{-1}\mathbf{N})^{-1}\mathbf{L}^{-1}, \quad (11)$$

and the existence of such an inverse depends on the existence of the two inverses indicated above. We may now observe the following.

- (i) It is fairly straightforward to comment on the existence of the inverse of a general linear operator \mathbf{L} .
- (ii) While it is less straightforward to comment in general on the existence of $(\mathbf{I} + \mathbf{L}^{-1}\mathbf{N})^{-1}$, we may note that the norm of the composite operator $\|\mathbf{L}^{-1}\mathbf{N}\|$ —(provided it exists) clearly a measure of the 'severity of nonlinearity' relative to the 'linearity'—will have a significant influence.
- (iii) More importantly, however, if these two inverses exist then the required nonlinear operator inverse \mathbf{P}^{-1} can be constructed and implemented as shown in Fig. 2, requiring only the inverse of the linear portion.

The advantages accruing if the nonlinear operator admits the representation in (10) are

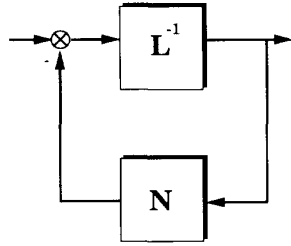


Fig. 2. Realization of partitioned nonlinear model inverse.

now clear: the nature of the operator inverse and the conditions for its existence are easier to determine; its construction and its implementation depend in a remarkably straightforward manner only on the linear subsystem inverse. We now note that the Volterra model not only naturally occurs in the partitioned form in (9), but its linear portion is precisely what conventional MPC is based upon. The rest of the paper is devoted to exploiting these facts for the development of a scheme to extend conventional linear model-based control schemes to nonlinear systems.

3. NONLINEAR SYSTEM DYNAMICS VIA VOLTERRA SERIES APPROXIMATIONS

3.1. Input/output representations in the time domain

For linear systems, the linear convolution integral

$$y(t) = \int_0^{\infty} h_1(\sigma)u(t-\sigma) d\sigma \quad (12)$$

(a functional that maps the entire past input to the value of the output at the present time) is well known as a useful model for representing arbitrary system dynamic behavior. Here $h_1(\sigma)$, the 'kernel' of the transformation, is the system's impulse response function. For nonlinear systems, Volterra generalized this functional representation in the form of a power series:

$$y(t) = y_1(t) + y_2(t) + y_3(t) + \dots, \quad (13)$$

where $y_1(t)$, the first-order term, is defined as in (12), and the other terms in the series are defined as follows:

$$y_2(t) = \int_0^{\infty} \int_0^{\infty} h_2(\sigma_1, \sigma_2)u(t-\sigma_1)u(t-\sigma_2) d\sigma_1 d\sigma_2 \quad (14)$$

$$y_3(t) = \int_0^{\infty} \int_0^{\infty} \int_0^{\infty} h_3(\sigma_1, \sigma_2, \sigma_3)u(t-\sigma_1) \times u(t-\sigma_2)u(t-\sigma_3) d\sigma_1 d\sigma_2 d\sigma_3 \quad (15)$$

$$y_i(t) = \int_0^{\infty} \dots \int_0^{\infty} h_i(\sigma_1, \dots, \sigma_i)u(t-\sigma_1) \dots u(t-\sigma_i) d\sigma_1 \dots d\sigma_i. \quad (16)$$

A discrete-time representation in the same spirit takes the form of the discrete power series

$$y(k) = y_1(k) + y_2(k) + y_3(k) + \dots, \quad (17)$$

where the first term, given by

$$y_1(k) = \sum_{i=1}^{\infty} h_1(i)u(k-i), \quad (18)$$

is recognizable as a variation of the linear convolution model employed in traditional MPC; higher-order terms are given by

$$y_2(k) = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} h_2(i, j)u(k-i)u(k-j), \quad (19)$$

$$y_3(k) = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \sum_{l=1}^{\infty} h_3(i, j, l)u(k-i)u(k-j)u(k-l). \quad (20)$$

The Volterra series representation is a temporal extension of the Taylor series expansion; it is thus limited to approximating systems with fading memory (Boyd and Chua 1985). Nevertheless, for a wide variety of nonlinear chemical processes that cannot be approximated adequately by the linear convolution model in (18) (e.g. exothermic chemical reactors or high-purity distillation columns), the Volterra series model, as a logical nonlinear extension of (12) or (18), often provides improved process representation.

These models may be represented compactly as

$$y = \mathbf{P}[u], \quad (21)$$

where \mathbf{P} is the nonlinear plant model operator that maps the input space to the output space. Observe from the model form that \mathbf{P} may be represented as

$$\mathbf{P} = \mathbf{P}_1 + \mathbf{P}_2 + \mathbf{P}_3 + \dots, \quad (22)$$

where, in the discrete case, the degree-1 (linear), degree-2 (quadratic) and degree-3 (cubic) operators are given respectively by

$$\mathbf{P}_1[u] = \sum_{i=1}^{\infty} h_1(i)u(k-i),$$

$$\mathbf{P}_2[u] = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} h_2(i, j)u(k-i)u(k-j),$$

$$\mathbf{P}_3[u] = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \sum_{l=1}^{\infty} h_3(i, j, l)u(k-i) \times u(k-j)u(k-l),$$

with the summations replaced by the appropriate

integral in the continuous case. In the following we shall make no distinction between operators representing discrete models and operators representing continuous models. The issue regarding the specific mathematical properties of the continuous, as compared with the discrete, versions of Volterra operators, and the implications they have for dynamic analysis in the continuous as compared with the discrete, time domain, have been carefully discussed by Schetzen (1980) and Rugh (1981). It is sufficient to note that in (21) and (22), as in the subsequent discussion, the operators are generalized and could be either continuous or discrete.

Observe now that, by separating the linear \mathbf{P}_1 from the rest of the higher-degree operators, referring to this residual sum collectively as \mathbf{N} , the Volterra model is clearly of the partition form in (9).

3.2. Obtaining Volterra series models

Volterra models may be obtained as follows.

- (i) From a nonlinear first principles model, by obtaining the Carleman linearization (approximation) (Schetzen, 1980; Rugh 1981) of the original system. The pertinent details of this procedure (which was actually employed in the example problems discussed in Section 5) can be found in the two cited references.
- (ii) From a polynomial NARMAX model, using a procedure due to Diaz and Desrochers (1988); the interested reader is referred to the cited reference for the details.
- (iii) From an artificial neural network. It has been shown recently (cf. Wray *et al.*, 1992) that for every artificial neural network that employs the usual sigmoidal squashing function, there is an exact—not approximate—equivalent infinite Volterra series representation. By using the infinite series representation of these sigmoidal functions, it is possible, through straightforward (but quite tedious) algebraic manipulations, to compute precisely the equivalent Volterra kernels from the synaptic weights. The interested reader is referred to Wray *et al.* (1992) for the complete details.
- (iv) From input/output data. A number of techniques have been proposed for computing the Volterra kernels strictly from input/output data gathered directly from the process. In each case the choice of input signals and the optimization criteria employed in computing parameter estimates are very critical elements of the identification procedure. New results and a detailed discussion of these and other related issues

are given by Pearson *et al.* (1992), to which the interested reader is referred. (cf. Seinfeld and Lapidus 1974; Schetzen, 1980; Rugh, 1981; Koh and Powers, 1985; Bendat, 1990; Pearson *et al.*, 1992).

4. CONTROLLER SYNTHESIS BASED ON VOLTERRA MODELS

4.1. Controller synthesis results

4.1.1. *Linear systems.* From Section 2, for the linear system modeled by

$$y = \mathbf{P}_1[u], \quad (23)$$

the basic controller synthesis result is

$$u = \mathbf{P}_1^{-1}[y_d - e]. \quad (24)$$

Even though, as previously noted, this may be modified as needed using the 'estimator' \mathbf{E} and the 'filter' \mathbf{F} , this controller is primarily based on \mathbf{P}_1^{-1} , the linear model inverse.

4.1.2. *Nonlinear systems.* Of the class of Volterra models, the simplest nonlinear extension of the linear convolution model in (23) is

$$y = (\mathbf{P}_1 + \mathbf{P}_2)[u], \quad (25)$$

the second-order Volterra model. From the result in Section 2, therefore, the controller synthesis procedure yields in this case

$$u = (\mathbf{I} + \mathbf{P}_1^{-1}\mathbf{P}_2)^{-1}\mathbf{P}_1^{-1}[\xi], \quad (26)$$

where

$$\xi = y_d - e. \quad (27)$$

The important point to note here is that such a controller can in fact be implemented as shown in Fig. 2: the 'main controller' still remains the linear model inverse; but it is now augmented with the auxiliary loop incorporating the quadratic nonlinear operator \mathbf{P}_2 .

It is now a straightforward exercise to show that in the general case, with

$$\mathbf{P} = \mathbf{P}_1 + \mathbf{P}_2 + \mathbf{P}_3 + \cdots, \quad (28)$$

the corresponding controller expression is

$$\mathbf{Q} = \mathbf{P}^{-1} = (\mathbf{I} + \mathbf{P}_1^{-1}\mathbf{P}_2 + \mathbf{P}_1^{-1}\mathbf{P}_3 + \cdots)^{-1}\mathbf{P}_1^{-1}, \quad (29)$$

and the corresponding block diagram will now have the residual nonlinear operators $(\mathbf{P}_2 + \mathbf{P}_3 + \cdots)$ in the auxiliary loop.

In the following subsection we derive the result that is presented in (26) using expansions of Volterra series operators (and their inverses). Once the equivalence has been established between the two approaches, we shall use the Volterra series expansion approach to derive controllers for broad classes of dynamical systems using generalized inverses.

4.2. The 'standard' approach

An alternative approach to the derivation of the controller in (26) is outlined below. Consider a controller operator \mathbf{Q} (to be determined) represented by

$$u = \mathbf{Q}[\xi], \quad (30)$$

with ξ as previously defined in (27). Substituting these expressions for \mathbf{Q} and ξ into (4) yields the control problem

$$\min_{\mathbf{Q}} \phi = \|\xi - \mathbf{P}[\mathbf{Q}[\xi]]\|. \quad (31)$$

This objective will clearly be minimized if the following relationship holds:

$$\mathbf{P}[\mathbf{Q}[\xi]] = \xi. \quad (32)$$

The discussions that follow will be very much concerned with the computation of the nonlinear operators \mathbf{P} and \mathbf{Q} appearing in (31) and (32). To simplify these discussions, define the nonlinear operator \mathbf{H} as this composition, i.e.

$$\mathbf{H} = \mathbf{P} * \mathbf{Q}, \quad (33)$$

where $*$ denotes composition of operators. As mentioned earlier, the resulting \mathbf{Q} will clearly be the inverse of \mathbf{P} (provided that it exists).

In following such a procedure, as noted in Chapter 7 of Schetzen (1980), it should be kept in mind that not all nonlinear systems possess such an inverse; also many possess an inverse only for a restricted range of input amplitude. One must therefore be careful to ensure that the range of the operator \mathbf{Q} is contained in the domain of the operator \mathbf{P} . This will be assumed throughout the following discussion. For further discussion of the various implications of this statement, see, for example, Chapter 7 of Schetzen (1980).

To obtain an explicit expression for \mathbf{Q} , using the operator notation defined above, with the plant represented by the operator

$$\mathbf{P} = \mathbf{P}_1 + \mathbf{P}_2 + \mathbf{P}_3 + \cdots, \quad (34)$$

the controller operator \mathbf{Q} is also set up as an infinite sequence of operators:

$$\mathbf{Q} = \mathbf{Q}_1 + \mathbf{Q}_2 + \mathbf{Q}_3 + \cdots, \quad (35)$$

with

$$\mathbf{Q}_1[\xi] = \sum_{i=1}^N v(i)\xi(k-i), \quad (36)$$

$$\mathbf{Q}_2[\xi] = \sum_{i=1}^N \sum_{j=1}^N w(i,j)\xi(k-i)\xi(k-j), \quad (37)$$

$$\mathbf{Q}_3[\xi] = \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N \psi(i,j,l)\xi(k-i)\xi(k-j)\xi(k-l) \quad (38)$$

in the discrete case, with the corresponding integral expressions in the continuous case.

Thus the controller synthesis problem requires solution of the equation

$$\mathbf{I} = \mathbf{P} * \mathbf{Q} = (\mathbf{P}_1 + \mathbf{P}_2 + \mathbf{P}_3 + \cdots) * (\mathbf{Q}_1 + \mathbf{Q}_2 + \mathbf{Q}_3 + \cdots) \quad (39)$$

for the controller \mathbf{Q} . By expanding the composition of the operators indicated in (39) and rearranging and grouping the results according to operator order and equating like terms, it has been shown (Schetzen, 1980) that the controller operator will be given explicitly by

$$\mathbf{Q}_1 = \mathbf{P}_1^{-1}, \quad (40)$$

$$\mathbf{Q}_2 = -\mathbf{P}_1^{-1} * \mathbf{P}_2 * \mathbf{Q}_1, \quad (41)$$

$$\mathbf{Q}_3 = -\mathbf{P}_1^{-1} * (\mathbf{P}_2 * (\mathbf{Q}_1 + \mathbf{Q}_2) - \mathbf{P}_2 * \mathbf{Q}_1 - \mathbf{P}_2 * \mathbf{Q}_2 + \mathbf{P}_3 * \mathbf{Q}_1), \quad (42)$$

$$\vdots \quad (43)$$

Note that this controller is an infinite sequence of operators, that the first term in the sequence is the linear model inverse, and that this is the only inverse operator present in the subsequent terms.

A key property of the expressions for the controller \mathbf{Q} as the Volterra operator inverse is the fact that the calculation of successive terms are completely decoupled. In other words, a linear operator inversion is all that is required initially, and subsequent terms depend only upon the same linear inverse, previous terms in the inverse operator series, and terms from the original plant operator series. However, there are three important issues that this approach does not address.

- (i) Since \mathbf{Q} is obtained by this technique as an infinite sequence of operators, it is pertinent to contemplate the issue of convergence. Under what conditions will the sequence in (35) converge?
- (ii) Even when the sequence can be shown to converge, how is the controller to be implemented in practice, particularly when the rate of convergence is slow?
- (iii) What happens when the inverse of \mathbf{P}_1 either does not exist or is unrealizable?

The procedure we have introduced in Section 4.1 resolves the first two issues, because

- (i) it provides an exact, analytical expression for the controller \mathbf{Q} represented as an infinite sequence using the alternative approach;
- (ii) as shown previously, it results in a controller that can be implemented as shown in the block diagram in Fig. 2.

This analytical representation for the controller

therefore has the dual advantage that it does not depend on the properties of an infinite series, and its implementation is straightforward and intuitive.

The third issue is resolved using generalized inverses, as we now discuss.

4.3. Controller synthesis using generalized inverses

In recognition of the fact that there will be situations in which \mathbf{P}_1 , the linear (degree-1) operator, either has no inverse or its inverse is undesirable for any number of reasons, we now proceed to develop explicit expressions for the controller in terms of appropriate generalized inverses.

4.3.1. Problem formulation and solution. In finding the exact inverse of \mathbf{P} and using it as the controller \mathbf{Q} , the strategy as presented in Section 4.2 has been to solve the series of equations resulting from (39). For our current purposes, we now reformulate the problem as follows. Since model prediction using \mathbf{P} and control action using \mathbf{Q} are respectively determined according to

$$y = \mathbf{P}[u], \quad (44)$$

$$u = \mathbf{Q}[\xi], \quad (45)$$

from the definition of each operator, we have first that

$$y = y^{(1)} + y^{(2)} + y^{(3)} + \dots, \quad (46)$$

where $y^{(i)}$ is the degree- i contribution to the overall signal y , arising from

$$y^{(i)} = \mathbf{P}_i[u]. \quad (47)$$

Similarly, we also have

$$u = u^{(1)} + u^{(2)} + u^{(3)} + \dots, \quad (48)$$

where $u^{(i)}$ is the degree- i contribution to the overall control action signal u , arising from

$$u^{(i)} = \mathbf{Q}_i[\xi]. \quad (49)$$

Finally, observe from (46) and (47) that

$$y = \mathbf{P}_1[u] + \mathbf{P}_2[u] + \mathbf{P}_3[u] + \dots, \quad (50)$$

but, from (48), u itself is the indicated infinite sequence. We now introduce the notation

$$\delta y^{(ij)} = \mathbf{P}_i[u^{(j)}], \quad (51)$$

with the first index denoting the degree of the operator and the second index the contributing degree of the component of the signal being operated upon. Thus, for example, $\delta y^{(12)}$ is that portion of $y^{(1)}$, the degree-1 contribution to y , due to the $u^{(2)}$, degree-2, component of the input signal u ; it is obtained by operating on $u^{(2)}$ by \mathbf{P}_1 .

Let r be the residual signal, the difference

between the plant/controller operator composition \mathbf{H} acting on ξ and the desired value of this composition (i.e. ξ itself):

$$r = \mathbf{H}[\xi] - \xi. \quad (52)$$

Now let r be defined in the same manner as y and u in (46) and (48), so that

$$r = r^{(1)} + r^{(2)} + r^{(3)} + \dots \quad (53)$$

That is, $r^{(i)}$ is given by

$$r^{(1)} = \mathbf{H}_1[\xi] - \xi, \quad (54)$$

$$r^{(j)} = \mathbf{H}_j[\xi] \quad \text{for } j = 2, 3, \dots, \quad (55)$$

where \mathbf{H}_j is the j th-order term in the expansion of the composition \mathbf{H} defined in (39). Observe now that obtaining the exact inverse of \mathbf{P} (demanding that $\mathbf{H} = \mathbf{I}$) is equivalent to requiring

$$r^{(i)} = 0 \quad \text{for all } i = 1, 2, 3, \dots \quad (56)$$

As shown previously, this requires the existence of \mathbf{P}_1^{-1} . It is possible to relax this requirement and, instead of (56), require only that the norm $\|r^{(i)}\|$ be minimized.

It is important to note that the required minimization is on each individual degree- i contributor to the residual signal, not the composite residual signal itself. What is required here is equivalent to minimizing the operator norm of the difference between corresponding degree- i operators, \mathbf{I}_i and \mathbf{H}_i , for each i . This clearly gives rise to a much stronger result, since it guarantees that each degree- i operator is 'close' as possible to the ideal operator of corresponding degree, and not merely that the overall operator \mathbf{H} is as 'close' to \mathbf{I} as possible.

Remarks. There are some subtle issues of mathematical rigor associated with the solution of the minimization problems we have posed.

- (i) First, there is the issue of the precise definition of the domain and range of the yet-undetermined, obviously nonlinear, operator \mathbf{Q} that we seek to derive via optimization. Discussions of this kind for the Volterra operator \mathbf{P} employed for the plant model may be found, for example in Section 1.5 of Rugh (1981). The converse discussion, in which it is recognized that in determining \mathbf{Q} as the inverse of \mathbf{P} , the domain of the latter is the range of the former, and vice versa, may be found in Chapter 7 of Schetzen (1980). It is relatively easy to show that the same arguments hold in this case, since what we are deriving is, in fact, a less stringent inverse of \mathbf{P} .
- (ii) A more serious issue has to do with the convergence of \mathbf{Q} . It is clear from the

earlier discussion that, in general, a stable, causal \mathbf{Q} does not exist as an exact inverse of \mathbf{P} if \mathbf{P}_1^{-1} is not stable or causal. However, as we shall soon show, a stable, causal \mathbf{Q} , which will depend on a generalized inverse for \mathbf{P}_1 , may be obtained under these circumstances. Such a \mathbf{Q} will of course be a pseudo-inverse of \mathbf{P} , but by choosing the criteria for pseudo-inverse carefully (see e.g. Ben-Israel and Greville, 1974), we can guarantee stability and causality for \mathbf{Q} . Nevertheless, even in this case, the resulting operator \mathbf{Q} , expressed as the Volterra series in (35), may converge only under a limited range of the amplitude of its input ξ . Further, as indicated in Section 7.5 of Schetzen (1980), this range is not easy to determine in general. These facts notwithstanding, we may make an appeal to the earlier discussion and employ the 'filter' \mathbf{F} and the estimator \mathbf{E} in the feedback loop as a means of ensuring that the amplitude of the feedback signal ξ is sufficiently modified for the purpose of achieving convergence for \mathbf{Q} .

- (iii) The notion of 'inversion' that is used here is fairly general, and can be relaxed, from a numerical perspective (Economou, 1985) or from a performance perspective, to yield robust controlled behavior. For instance, the analytical solution to the standard unconstrained model predictive control problem contains an explicit weighting term that relaxes the explicit model inversion as a direct consequence of incorporating a penalty on the manipulated variable. For a nice summary of this notion of relaxed model inversion, see Hernández (1992).
- (iv) Finally, even though we shall soon show that the solution of the specific optimization problem posed above technically involves only the linear operator \mathbf{P}_1 explicitly, it is nevertheless also clear from (52) that at the foundation of the entire exercise is the notion of the norm of nonlinear operators (for $i > 1$). While the notion of the norm of a linear operator is straightforward and well known, it is not as obvious or as straightforward with nonlinear operators. It is clearly outside the intended scope of this paper to provide even a cursory discussion of the precise mathematical foundations underlying the operations indicated in (52) and which make it possible to derive from it the results we shall soon present. It is sufficient to note that most of the pertinent issues are taken up with enough detail for example in Chapter 2 of Berger (1977). The

specific issues regarding the spaces of nonlinear operators, as well as those regarding the existence of solutions to nonlinear operator equations in Banach spaces (such as the kind we have posed in (52)) are considered in detail in Chapters 3 and 4 of Martin 1976. The interested reader is encouraged to consult these references.

We now proceed to obtain the solution to the optimization problem.

Degree 1. In this case, since

$$\mathbf{H}_1[\xi] = \mathbf{P}_1 * \mathbf{Q}_1[\xi], \quad (57)$$

and $\mathbf{I}_1[\xi] = \xi$, we have

$$r^{(1)} = \mathbf{P}_1 * \mathbf{Q}_1[\xi] - \xi, \quad (58)$$

but, by definition, $\mathbf{Q}_1[\xi] = u^{(1)}$, so that the optimization problem becomes

$$\|r^{(1)}\| = \|\mathbf{P}_1[u^{(1)}] - \xi\|. \quad (59)$$

If in particular $\|\cdot\|$ refers to the L_2 norm then (59) has the well-known solution

$$u^{(1)} = \mathbf{P}_1^\dagger[\xi], \quad (60)$$

where the operator \mathbf{P}_1^\dagger is the left inverse of \mathbf{P}_1 defined by

$$\mathbf{P}_1^\dagger \mathbf{P}_1 = \mathbf{I}_1. \quad (61)$$

Of course, since, by definition, $u^{(1)} = \mathbf{Q}_1[\xi]$, it follows from (60) that the linear portion of the \mathbf{Q} operator is given by

$$\mathbf{Q}_1 = \mathbf{P}_1^\dagger \quad (62)$$

(compare with $\mathbf{Q}_1 = \mathbf{P}_1^{-1}$ for the exact inverse controller).

Degree 2. In this case

$$\mathbf{H}_2[\xi] = \mathbf{P}_1 * \mathbf{Q}_2[\xi] + \mathbf{P}_2 * \mathbf{Q}_1[\xi], \quad (63)$$

or, equivalently,

$$\mathbf{H}_2[\xi] = \mathbf{P}_1[u^{(2)}] + \mathbf{P}_2[u^{(1)}]. \quad (64)$$

If we keep in mind that, from (60), $u^{(1)}$ is now a known quantity, and recall the definition of $\delta y^{(ii)}$, we observe immediately that the expression in (64) becomes

$$\mathbf{H}_2[\xi] = \mathbf{P}_1[u^{(2)}] + \delta y^{(21)}; \quad (65)$$

and finally, since $\mathbf{I}_2[\xi] = 0$, we then have the second-degree optimization problem as

$$\|r^{(2)}\| = \|\mathbf{P}_1[u^{(2)}] + \delta y^{(21)}\|, \quad (66)$$

a problem entirely similar in structure to (59), and involving only the linear operator \mathbf{P}_1 , even though the input signal in question, $u^{(2)}$, is a second-degree contribution. The solution is

$$u^{(2)} = -\mathbf{P}_1^\dagger[\delta y^{(21)}], \quad (67)$$

with the same \mathbf{P}_1^\dagger as in (60).

Degree 3. Here the problem is somewhat more complicated, but the principles are essentially the same. In this case

$$\mathbf{H}_3[\xi] = \mathbf{P}_1 * \mathbf{Q}_3[\xi] + \mathbf{P}_3 * \mathbf{Q}_1[\xi] - \mathbf{P}_2 * \mathbf{Q}_1[\xi] - \mathbf{P}_2 * \mathbf{Q}_2[\xi] + \mathbf{P}_2 * [\mathbf{Q}_1[\xi] + \mathbf{Q}_2[\xi]], \quad (68)$$

or, equivalently,

$$\mathbf{H}_3[\xi] = \mathbf{P}_1[u^{(3)}] + \mathbf{P}_3[u^{(1)}] - \mathbf{P}_2[u^{(1)}] - \mathbf{P}_2[u^{(2)}] + \mathbf{P}_2[u^{(1)} + u^{(2)}]. \quad (69)$$

Since, by now, both $u^{(1)}$ and $u^{(2)}$ are known quantities, we may simplify (69) by defining

$$\delta y^{(3)} = \delta y^{(31)} - \delta y^{(21)} - \delta y^{(22)} + \delta y^{(2S_{12})}, \quad (70)$$

with $\delta y^{(2S_{12})}$ defined as

$$\delta y^{(2S_{12})} = \mathbf{P}_2[u^{(1)} + u^{(2)}] \quad (71)$$

and the others as previously defined; the result is that (69) becomes

$$\mathbf{H}_3[\xi] = \mathbf{P}_1[u^{(3)}] + \delta y^{(3)}, \quad (72)$$

and again, since $\mathbf{I}_3[\xi] = 0$, we have the third-degree optimization problem as

$$\|r^{(3)}\| = \|\mathbf{P}_1[u^{(3)}] + \delta y^{(3)}\|. \quad (73)$$

Observe again that this is a problem entirely similar in structure to (59), involving again only the linear operator \mathbf{P}_1 . (The input signal in question, $u^{(3)}$, of course, is a third-degree contribution). The solution again is

$$u^{(3)} = -\mathbf{P}_1^+[\delta y^{(3)}], \quad (74)$$

with the same \mathbf{P}_1^+ as in the degree-1 and -2 problems.

It is now possible to generalize: the degree- i problem will always have the form

$$\|r^{(i)}\| = \|\mathbf{P}_1[u^{(i)}] + \delta y^{(i)}\|, \quad (75)$$

where $\delta y^{(i)}$ consists only of known quantities dependent on $u^{(i-1)}$, and other components of lower degree up to and including $u^{(1)}$.

We now observe, once again, that, as previously obtained under the assumption that \mathbf{P}_1^{-1} exists, the controller implied above is an infinite sequence, with each term involving only \mathbf{P}_1^+ this time. It can be shown that the corresponding analytical expression for this infinite sequence controller is precisely as in (26) and (29), but now with \mathbf{P}_1^{-1} replaced with \mathbf{P}_1^+ .

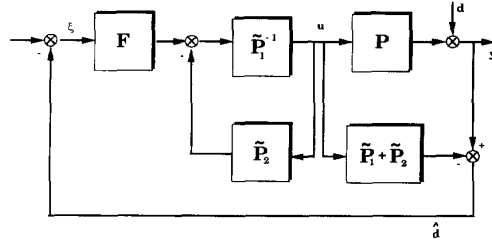


Fig. 3. Structure for second-order Volterra controller.

4.3.2. Implications for constrained discrete MPC. The fact that the problem of generating the controller \mathbf{Q} can be thus posed as an optimization problem has some significant implications for controller implementation within the explicit MPC framework. Observe that the norms to be minimized could be weighted, and the weights could in fact be made dependent on the operator degree; constraints could also be introduced (in the multivariable case) if needed. Furthermore, the entire controller could be implemented as in Fig. 3, with \mathbf{P}_1^{-1} replaced by the linear MPC controller, noting the crucial point that the auxiliary loop endows the otherwise well-known linear MPC controller with nonlinear characteristics by providing a form of ‘correction’ for the effect of nonlinearities.

These and other related issues are significant enough in their own right that they warrant a full-scale discussion, which we shall defer to a subsequent paper.

5. CASE STUDY

5.1. Continuous polymerization reactor example

In this section we consider the application of the proposed controller design strategy to a polymerization reaction taking place in a jacketed CSTR. The reaction under consideration is the free-radical polymerization of methyl methacrylate (MMA) with azo-bis-isobutyronitrile (AIBN) as initiator and toluene as solvent. Details of the rate laws for free-radical polymerization were obtained from Congalidis *et al.* (1989), Daoutidis *et al.* (1990) and Ray (1972). For this particular case study, we make the following simplifying assumptions:

- isothermal operation;
- perfect mixing;
- constant heat capacity;
- no polymer in the inlet stream;
- no gel effect;
- constant reactor volume;
- negligible initiator flow rate (in comparison with monomer flow rate);

• quasisteady state and long-chain hypothesis.
Under these assumptions, the following model is obtained for the polymerization reactor:

$$\begin{aligned}
 \dot{x}_1 &= - \left[Z_p \exp \left(\frac{-E_p}{RT} \right) \right. \\
 &\quad \left. + Z_{t_m} \exp \left(\frac{-E_{t_m}}{RT} \right) \right] x_1 P_0(x_2, T) - \frac{F x_1}{V} + \frac{F C_{m_0}}{V}, \\
 \dot{x}_2 &= - Z_i \exp \left(\frac{-E_i}{RT} \right) x_2 - \frac{F x_2}{V} + \frac{F_i C_{i_0}}{V}, \\
 \dot{x}_3 &= \left[0.5 Z_{T_c} \exp \left(\frac{-E_{T_c}}{RT} \right) \right. \\
 &\quad \left. + Z_{T_d} \exp \left(\frac{-E_{T_d}}{RT} \right) \right] P_0^2(x_2, T) \\
 &\quad + Z_{t_m} \exp \left(\frac{-E_{t_m}}{RT} \right) x_1 P_0(x_2, T) - \frac{F x_3}{V}, \\
 \dot{x}_4 &= M_m \left[Z_p \exp \left(\frac{-E_p}{RT} \right) \right. \\
 &\quad \left. + Z_{t_m} \exp \left(\frac{-E_{t_m}}{RT} \right) \right] x_1 P_0(x_2, T) - \frac{F x_4}{V},
 \end{aligned} \tag{76}$$

where

$$\begin{aligned}
 P_0(x_2, T) &= \left[\frac{2 f^* x_2 Z_i \exp(-E_i/RT)}{Z_{T_d} \exp(-E_{T_d}/RT) + Z_{T_c} \exp(-E_{T_c}/RT)} \right]^{1/2}.
 \end{aligned}$$

The dimensionless state variables in the above equations are defined as follows: $x_1 = C_m$, $x_2 = C_i$, $x_3 = D_0$, $x_4 = D_1$, where C_m is the monomer concentration, C_i is the initiator concentration, and D_1/D_0 is the number-average molecular weight. The control problem focuses on manipulating the volumetric flowrate of the initiator ($u = F_i$) in order to regulate the number-average molecular weight ($y = D_1/D_0$). The operating conditions chosen for this study are as follows: $x_{10} = 5.50677$, $x_{20} = 0.132906$, $x_{30} = 0.0019752$, $x_{40} = 49.3818$, $u_0 = 0.016783$, $y_0 = 25\,000.5$.

Introducing the values in Tables 1 and 2 for the physical constants in the modeling equations (and assuming a constant C_{m_0}) yields:

$$\begin{aligned}
 \dot{x}_1 &= 10(6 - x_1) - 2.4568 x_1 \sqrt{x_2}, \\
 \dot{x}_2 &= 80u - 10.1022 x_2, \\
 \dot{x}_3 &= 0.0024121 x_1 \sqrt{x_2} + 0.112191 x_2 - 10 x_3, \\
 \dot{x}_4 &= 245.978 x_1 \sqrt{x_2} - 10 x_4, \\
 y &= \frac{x_4}{x_3}.
 \end{aligned} \tag{77}$$

Table 1. Kinetic parameters

i	Z_i	E_i
T_c	$3.8223 \times 10^{10} \text{ m}^3 \text{ kmol}^{-1} \text{ h}^{-1}$	$2.9442 \times 10^3 \text{ kJ kmol}^{-1}$
T_d	$3.1457 \times 10^{11} \text{ m}^3 \text{ kmol}^{-1} \text{ h}^{-1}$	$2.9442 \times 10^3 \text{ kJ kmol}^{-1}$
I	$3.7920 \times 10^{18} \text{ h}^{-1}$	$1.2550 \times 10^5 \text{ kJ kmol}^{-1}$
P	$1.7700 \times 10^9 \text{ m}^3 \text{ kmol}^{-1} \text{ h}^{-1}$	$1.8283 \times 10^4 \text{ kJ kmol}^{-1}$
f_m	$1.0067 \times 10^{15} \text{ m}^3 \text{ kmol}^{-1} \text{ h}^{-1}$	$7.4478 \times 10^4 \text{ kJ kmol}^{-1}$
$f^* = 0.58$		

Table 2. System parameters

$F = 1.00 \text{ m}^3$
$V = 0.1 \text{ m}^3$
$C_{i_0} = 8.0 \text{ kmol m}^{-3}$
$y^{sp} = 25\,000.5 \text{ kg kmol}^{-1}$
$F_i = 0.016783 \text{ m}^3 \text{ h}^{-1}$
$R = 8.314 \text{ kJ kmol}^{-1} \text{ K}^{-1}$
$M_m = 100.12 \text{ kg kmol}^{-1}$
$C_{m_0} = 6.0 \text{ kmol m}^{-3}$
$T = 335 \text{ K}$

From this first-principles reactor model, we now proceed to obtain an approximate Volterra model via the Carlemann linearization approach. By defining the normalized variables $z_i = (x_i - x_{i0})/x_{i0}$, where x_{i0} is the nominal operating value for x_i , we obtain a model with variables that are zero at the operating condition and have roughly equivalent ranges. Now, if we take a Taylor series approximation of this normalized model, and retain terms up to second order, we obtain

$$\begin{aligned}
 \dot{z}_1 &= -10.8957 z_1 - 0.447837 z_2 - 0.447837 z_1 z_2 \\
 &\quad + 0.111959 z_2^2, \\
 \dot{z}_2 &= -10.1022 z_2 + 10.1022 \bar{u}, \\
 \dot{z}_3 &= 2.45162 z_1 + 8.7744 z_2 - 10 z_3 \\
 &\quad + 1.22581 z_1 z_2 - 0.306453 z_2^2, \\
 \dot{z}_4 &= 10 z_1 + 5.00001 z_2 - 10 z_4 \\
 &\quad + 5.00001 z_1 z_2 - 1.25 z_2^2, \\
 \dot{\bar{y}} &= -z_3 + z_4 - z_3 z_4 + z_3^2.
 \end{aligned}$$

The Carlemann linearization of this nonlinear system yields the following bilinear approximation, where

$$\begin{aligned}
 \tilde{z} &= [z_1 \quad z_2 \quad z_3 \quad z_4 \quad z_1^2 \quad z_1 z_2 \quad z_1 z_3 \quad z_1 z_4 \\
 &\quad z_2^2 \quad z_2 z_3 \quad z_2 z_4 \quad z_3^2 \quad z_3 z_4]^T \\
 \dot{\tilde{z}} &= A \tilde{z} + N \tilde{z} u + b u, \\
 \bar{y} &= c^T \tilde{z}.
 \end{aligned} \tag{78}$$

The matrices in this equation are given by:

$$A = \begin{bmatrix} -10.8957 & -0.447837 & 0 & 0 & 0 & -0.447837 & 0 & 0 & 0.111959 & 0 & 0 & 0 & 0 \\ 0 & -10.1022 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2.45162 & 8.7744 & -10 & 0 & 0 & 1.22581 & 0 & 0 & -0.306453 & 0 & 0 & 0 & 0 \\ 10 & 5.00001 & 0 & -10 & 5.00001 & 0 & 0 & 0 & -1.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -21.7913 & -0.895674 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -20.9979 & 0 & 0 & -0.447837 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.45162 & 8.7744 & -20.8957 & 0 & 0 & -0.447837 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 5.00001 & 0 & -20.8957 & 0 & 0 & -0.447837 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -20.2045 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.45162 & 0 & 0 & 8.7744 & -20.1022 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 5.00001 & 0 & -20.1022 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4.90325 & 0 & 0 & 17.5488 & 0 & -20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 & 2.45162 & 0 & 5.00001 & 8.7744 & 0 & -20 \end{bmatrix},$$

$$N = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10.1022 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 20.2044 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10.1022 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10.1022 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$b = [0 \quad 10.1022 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0],$$

$$c^T = [0 \quad 0 \quad -1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad -1].$$

5.1.1. *Controller synthesis.* From the bilinear model of the polymerization reactor, it is relatively straightforward to derive the linear process controller as well as the second-order correction described in Section 3.

Using the matrices given above, the linear reactor model can be described by the following frequency-domain Volterra kernel (Laplace transfer function):

$$P_1(s) = c_1^T(sI - A_{11})^{-1}b_1, \quad (79)$$

where A_{11} , b_1 , c_1 is the subsystem of the bilinear model corresponding to the first four (linear) states in \bar{z} . The second-order portion of the model can be described by the frequency-domain Volterra kernels

$$P_2(s_1, s_2) = c^T[(s_1 + s_2)I - A]^{-1}N(s_1 - A)^{-1}b, \quad (80)$$

which can be realized by the following expression in the time domain (Al-Baiyat, 1986):

$$\begin{aligned} \dot{x}_1 &= Ax_1 + bu, \\ \dot{x}_2 &= Ax_2 + N(x_1 * u), \end{aligned} \quad (81)$$

$$y = cx_2. \quad (82)$$

From this information, a standard controller can be designed for P_1 (denoted by Q_1) using any appropriate method. In this study we employ an IMC controller where Q_1 is chosen as the inverse

of P_1 , and this is augmented by a linear filter

$$F(s) = \frac{1}{(0.5s + 1)^2} \quad (83)$$

to yield the causal controller

$$\frac{-0.1056s^4 - 4.324s^3 - 66.33s^2 - 452.0s - 1155}{s^4 + 25.79s^3 + 209.1s^2 + 588s + 471.6}. \quad (84)$$

Following the procedure outlined in Section 3, a second-order controller is realized as shown in Fig. 3. Note that the linear filter F must be implemented before the second-order correction is fed back. Structurally, this is equivalent to implementing a controller as in Fig. 2 with the linear plant inverse augmented with a linear filter, and the second-order correction P_2 augmented with the inverse of the linear filter. The causality of this latter operator is not strictly guaranteed, since the relative degree or pole-zero excess of the second-order Volterra kernel is not always equal to the relative degree of the linear Volterra kernel. It is always possible to correct for this by introducing a second linear filter F_2 , cascaded with the second-order plant model term. Further discussion of this topic is postponed.

Observe that the controller has been synthesized using only an optimal linear controller (generated from the linear inverse)

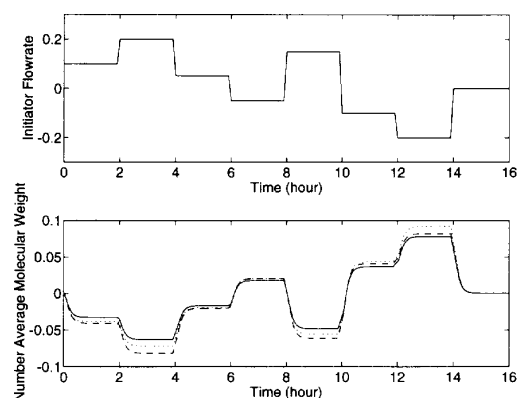


Fig. 4. Open-loop simulations of polymerizer reactor for step inputs of small magnitude.

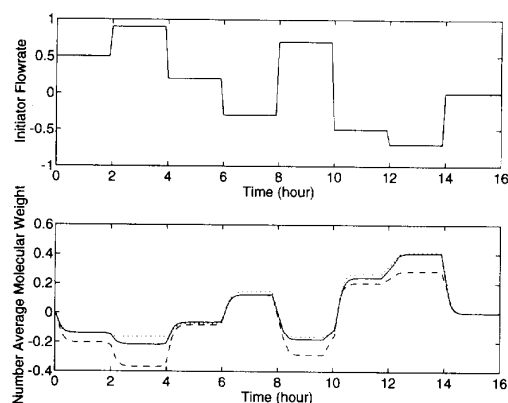


Fig. 5. Open-loop simulations of polymerizer reactor for step inputs of large magnitude.

and the second-order Volterra kernel for the reactor model.

5.1.2. Simulation results. The nonlinear nature of the polymerization reactor under investigation is revealed in the open-loop simulations in Figs 4 and 5. Figure 4 shows the response of the full nonlinear model (solid line, (77)), the second-order bilinear (Volterra) approximation (dotted line, (78)) and the linear approximation (dashed line, (79)), to step changes in the initiator flow rate that range $\pm 20\%$. As can be seen in the figure, all three curves overlap for small input changes, but as the magnitude of the change increases, the nonlinear curve separates from the other two. However, the second-order Volterra model tracks the true nonlinear response more accurately than the linear model.

Results from a similar simulation are displayed in Fig. 5, where now the magnitude of the input step change has been increased to $\pm 90\%$. While these may appear to be unrealistic changes, they do highlight the relative accuracy of the linear and second-order model approximations. In this case there is a significantly larger difference between the true nonlinear response and both the linear and second-order models. And, as before, the second-order response tracks the full nonlinear response more closely than the linear response.

Figure 6 shows the response of the closed-loop system to a -10% step change in the set-point:

- (i) closed-loop with linear controller (dashed line);
- (ii) Closed-loop with linear controller plus second-order correction (Volterra controller) (dotted line);

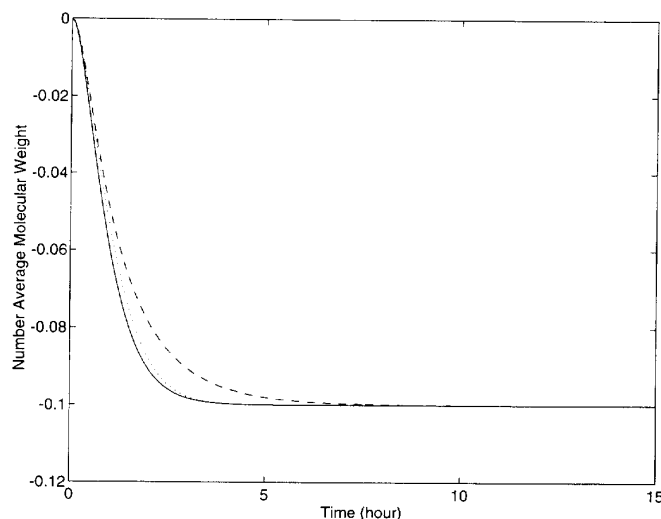


Fig. 6. Closed-loop simulation of polymerizer reactor for a -10% step in the set-point.

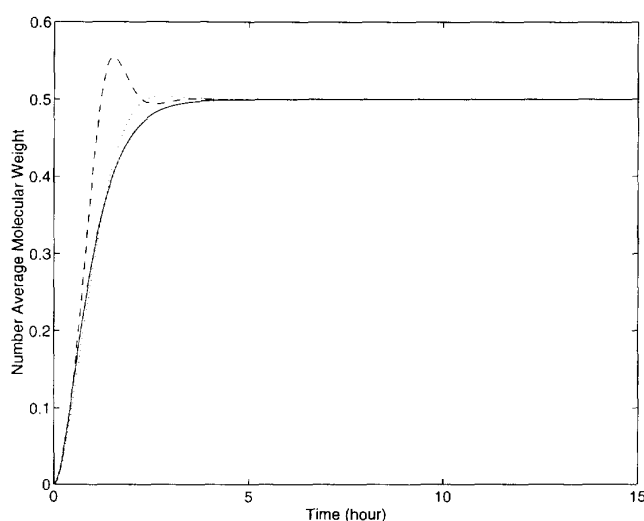


Fig. 7. Closed-loop simulation of polymerizer reactor for a +50% step in the set-point.

(iii) Reference trajectory (linear filter response) (solid line).

For this set-point change, the nonlinear nature of the plant is revealed as the closed-loop response deviates from the ideal linear trajectory. While both controllers produce similar responses, the second-order controller has reduced the integral squared tracking error, relative to the linear controller, by approximately 95%.

The same simulation is repeated for a larger step change ($r=0.5/s$), and the results are displayed in Fig. 7. The same three curves are plotted (linear controller dashed, second-order controller dotted, and reference trajectory solid), and similar trends are evident. In this case the larger input magnitude has led to a greater separation between the linear and second-order response, with the latter tracking the reference trajectory more closely.

A quantitative summary of the response to a number of set-point changes for three different reactor plant descriptions is given in Table 3, which shows the magnitudes of the integral squared error between the true output and the reference trajectory. In this case the reactor plant employed in the simulation is the full nonlinear plant model given in (77).

Table 3. Closed-loop performance—nominal nonlinear plant

Input signal	Tracking error	
	Linear controller	Volterra controller
$r = 0.1/s$	6.7759×10^{-5}	0.1398×10^{-5}
$r = 0.25/s$	2.4136×10^{-3}	0.1988×10^{-3}
$r = 0.5/s$	2.6864×10^{-2}	2.1912×10^{-2}
$r = -0.05/s$	4.6581×10^{-6}	0.0457×10^{-6}
$r = -0.1/s$	7.7548×10^{-5}	0.4076×10^{-5}
$r = -0.15/s$	4.0764×10^{-4}	0.7313×10^{-4}

Table 4. Closed-loop performance—perturbed nonlinear plant

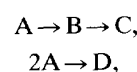
Input signal	Tracking error	
	Linear controller	Volterra controller
$r = 0.1/s$	1.0566×10^{-5}	9.3754×10^{-5}
$r = 0.25/s$	8.2126×10^{-4}	3.0921×10^{-4}
$r = 0.5/s$	2.0392×10^{-2}	0.1098×10^{-2}
$r = -0.1/s$	4.7520×10^{-5}	1.7597×10^{-5}
$r = -0.25/s$	3.3347×10^{-4}	0.3865×10^{-4}
$r = -0.5/s$	1.1838×10^{-3}	0.0071×10^{-3}

Another set of informative simulations is summarized in Table 4. In this case we examine the robust performance of the two closed-loop systems (linear controller and second-order controller). The specific uncertainty considered was a -15% error in the physical parameter corresponding to the gain coefficient on the manipulated variable. As before, the integral squared error from a reference trajectory is compiled. Although a slight performance advantage for the linear controller is observed for the smallest positive input, the remaining results confirm the superior performance of the second-order controller under a situation of nonlinear model mismatch.

In both cases considered above (nominal nonlinear plant and perturbed nonlinear plant) it can be seen that the second-order controller consistently outperforms the linear controller by a margin of approximately 20–95%.

5.2. Van de Vusse reactor example

The second application considered is the van de Vusse reaction kinetic scheme:



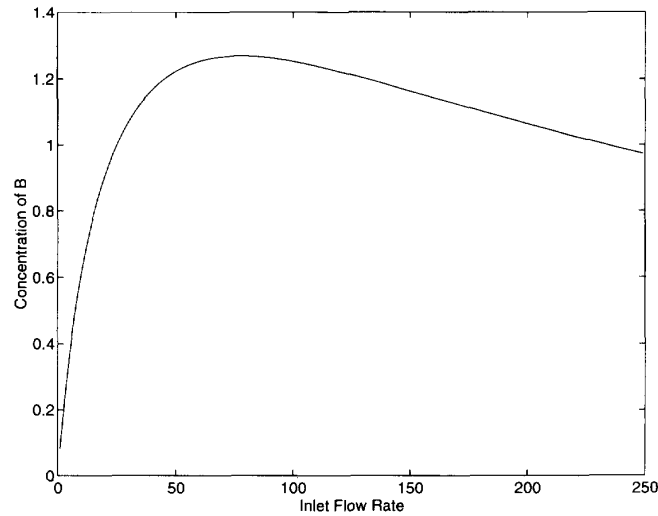


Fig. 8. Operating locus of van de Vusse reactor.

which is carried out in an isothermal CSTR. The mass balances for components A and B are given by

$$\dot{C}_A = -k_1 C_A - k_3 C_A^2 + \frac{F}{V} (C_{Af} - C_A),$$

$$\dot{C}_B = k_1 C_A - k_2 C_B - \frac{F}{V} C_B,$$

$$y = C_B.$$

The control problem focuses on regulating the concentration of component B by manipulating the inlet flow rate. This example has been considered by a number of researchers as a benchmark problem for nonlinear process control algorithms (van de Vusse, 1964; Kantor, 1986; Doyle, 1991). A plot of the system's operating locus (Fig. 8) reveals some of the interesting behavior displayed by this simple system. The reactor exhibits a change in gain at peak conversion level, and displays non-minimum-phase behavior for operation to the left of this peak and minimum-phase behavior for operating points on the right.

By normalizing the process variables about the operating point

$$C_{A0} = 3.0,$$

$$C_{B0} = 1.12,$$

$$\frac{F_0}{V} = 34.3$$

and substituting the values for the physical constants (Table 5), the process model becomes

$$\begin{aligned} \dot{x}_1 &= -50x_1 - 10x_1^2 + u(10 - x_1), \\ \dot{x}_2 &= 50x_1 - 100x_2 + u(-x_2), \\ y &= x_2, \end{aligned} \quad (85)$$

where the deviation variable for the concentration of component A is denoted by x_1 , the

Table 5. Kinetic parameters

k_1	50 h ⁻¹
k_2	100 h ⁻¹
k_3	10 l mol ⁻¹ h ⁻¹
C_{Af}	10 mol l ⁻¹
V	1 L

concentration of component B by x_2 and the inlet flow rate by u .

The Carleman linearization of this nonlinear model yields the following bilinear approximation, where $\tilde{z} = [z_1 \ z_2 \ z_1 z_2 \ z_1^2 \ z_2^2]^T$:

$$\dot{\tilde{z}} = A\tilde{z} + N\tilde{z}u + bu,$$

$$\tilde{y} = c^T \tilde{z}.$$

The matrices in this equation are given by

$$A = \begin{bmatrix} -144.3 & 0 & 0 & -10 & 0 \\ 50 & -134.3 & 0 & 0 & 0 \\ 0 & 0 & -278.6 & 50 & 0 \\ 0 & 0 & 0 & -288.6 & 0 \\ 0 & 0 & 100 & 0 & -268.6 \end{bmatrix},$$

$$N = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ -1.12 & 7 & -2 & 0 & 0 \\ 14 & 0 & 0 & -2 & 0 \\ 0 & -2.24 & 0 & 0 & -2 \end{bmatrix},$$

$$b = \begin{bmatrix} 7 \\ -1.12 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$c^T = [0 \ 1 \ 0 \ 0 \ 0]$$

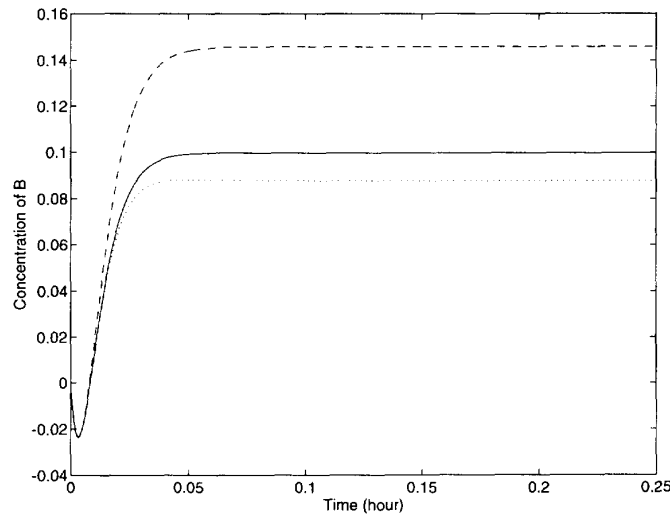


Fig. 9. Open-loop simulations of van de Vusse reactor for positive step input.

The control synthesis procedure from Section 5.1 is repeated, using this bilinear approximation. An IMC controller is employed, with the following realization of the plant inverse augmented with a first-order filter (time constant 0.01 h):

$$Q_1 = \frac{89.28s^2 + 24\,880s + 1.730 \times 10^6}{s^2 + 268.2s + 16\,820}. \quad (86)$$

5.2.1. Simulation results. Figures 9 and 10 show the open-loop response of the three different system models to step changes in the inlet flow rate of +15 and -20 respectively. The three models (linear dashed, second-order Volterra dotted and original nonlinear solid)

show a range of responses for the concentration of component B; however, the second-order Volterra model shows a closer correspondence to the true nonlinear response than does the linear response model. All three models display the inverse associated with a non-minimum-phase system, and all three display the same sign of the process gain.

As before, a linear model-based control scheme is to be compared with one based on a second-order Volterra model. In this case, however, a decomposition of the original linear system approximation into all-pass and minimum-phase factors must first be done. The linear controller is designed for the minimum-phase factor, leaving the all-pass factor and the filter in the nominal design.

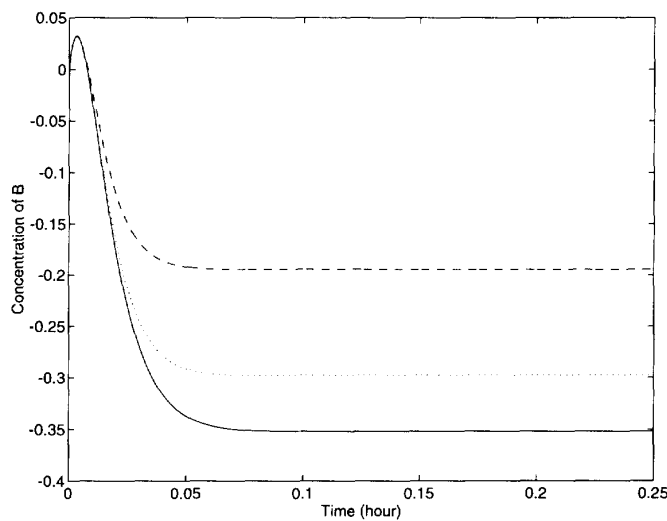


Fig. 10. Open-loop simulations of van de Vusse reactor for negative step input.

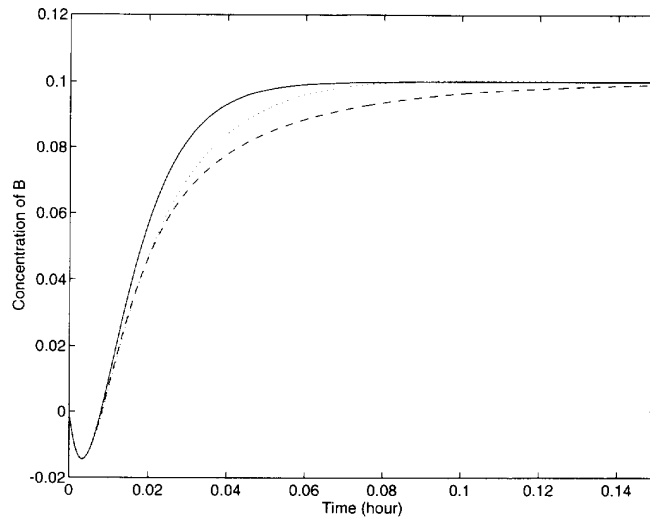


Fig. 11. Closed-loop simulation of van de Vusse reactor for a +10% step in the set-point.

Figure 11 shows the closed-loop response of the system to a 0.1 step increase in the set-point of C_B . The solid line is the ideal system response, in other words, the response of the product of the linear all-pass factor and the first order filter to a step input. The dashed line is the linear model-based controlled system and the dotted line is the second-order Volterra model-based scheme. It is clear that the Volterra model-based system shows a closer correspondence to the reference signal, while the linear model-based system shows a more pronounced sluggish response.

Figure 12 shows the closed-loop response to a -0.5 change in the set-point of C_B , further highlighting the difference between the two

controllers. The Volterra model-based controller shows slight undershoot before returning quickly to the new set-point; the linear model-based system shows greater undershoot and a slowly oscillating return to the set-point.

This second example demonstrates two key points:

- The Volterra model-based control scheme can directly handle non-minimum-phase systems through appropriate design of the linear portion of the controller.
- This van de Vusse reactor represents a more nonlinear dynamical system than the previous polymerizer example, and the relative strength of the Volterra model-based scheme is highlighted.

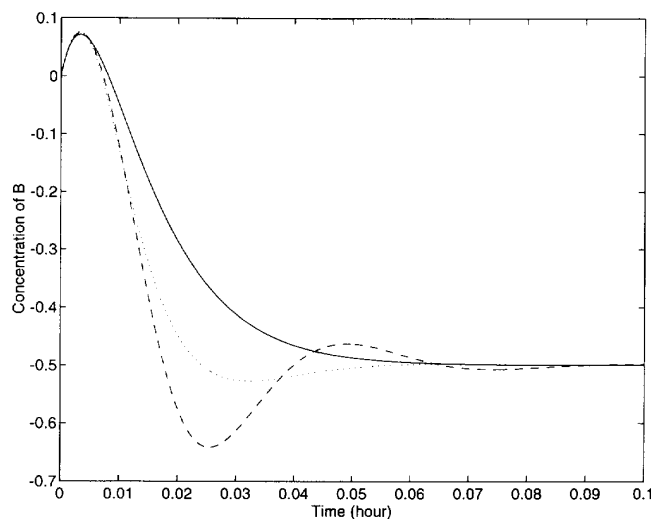


Fig. 12. Closed-loop simulation of van de Vusse reactor for a -50% step in the set-point.

6. SUMMARY

In this paper we have presented a systematic procedure for nonlinear model-based controller design using Volterra models. These models are similar in structure to the conventional MPC models, and can be obtained from input/output data; they therefore provide a means for preserving the original spirit of conventional MPC in tackling nonlinear problems. The controller can be implemented as a linear controller augmented with an auxiliary loop incorporating corrections for the effect of nonlinearities. Thus, the computational requirements for this procedure are minimal: (i) a second-order Volterra model and (ii) a linear controller derived from the inverse (or pseudo-inverse) of the linear portion of the process model.

The performance of the nonlinear model-based controller was evaluated using two examples.

- For a moderately nonlinear, but practical, polymerization reactor, it was shown that the second-order correction provided significant performance improvement over a wide range of conditions.
- For a highly nonlinear benchmark problem, the van de Vusse reaction scheme, it was demonstrated that the performance of the Volterra model-based controller was significantly better than that of a linear controller. This example also demonstrated the ability of this approach to handle non-minimum-phase systems.

Current efforts in this area involve explicit MPC implementation of this Volterra model-based controller, particularly for multivariable systems with constraints. Another direction currently under investigation is the integration of the identification results described by Pearson *et al.* (1992) with the controller synthesis techniques described in this paper to yield a unified approach to nonlinear process identification and control. These results will be discussed in subsequent papers.

REFERENCES

- Al-Baiyat, S. (1986). Nonlinear feedback synthesis: a Volterra approach. PhD thesis, Electrical and Computer Engineering Department, University of Notre Dame.
- Al-Baiyat, S. and Sain, M. (1986). Control design with transfer functions associated to higher order Volterra kernels. In *Proc. 25th IEEE Conf. on Decision and Control*, pp. 1306–1311.
- Bartee, J. and Georgakis, C. (1992). Identification and control of bilinear systems. In *Proc. American Control Conf.*
- Ben-Israel, A. and T. Greville (1974). *Generalized Inverses: Theory and Applications*. Krieger, New York.
- Bendat, J. (1990). *Nonlinear System Analysis and Identification from Random Data*. Wiley, New York.
- Berger, M. (1977). *Nonlinearity and Functional Analysis*. Academic Press, New York.
- Biegler, L. and J. Rawlings (1991). Optimization approaches to nonlinear model predictive control. In *Proc. 4th International Conf. of Chemical Process Control*.
- Boyd, S. and L. Chua (1985). Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Trans. Circuits Syst.*, **CAS-32**, 1150–1161.
- Congalidis, J., J. Richards and W. Ray (1989). Feedforward and feedback control of a copolymerization reactor. *AIChE J.*, **35**, 891–907.
- Daoutidis, P., M. Sorousch and C. Kravaris (1990). Feedforward/feedback control of multivariable nonlinear processes. *AIChE J.*, **36**, 1471–1484.
- Diaz, H. and A. Desrochers (1988). Modeling of nonlinear discrete-time systems from input–output data. *Automatica*, **24**, 629–641.
- Doyle III, F. (1991). Robustness properties of nonlinear process control and implications for the design and control of a packed bed reactor. PhD thesis, Chemical Engineering Department, California Institute of Technology.
- Eaton, J. and J. Rawlings (1990). Feedback control of chemical processes using on-line optimization techniques. *Comput. Chem. Engng.*, **14**, 469–479.
- Economou, C. (1985). An operator theory approach to nonlinear controller design. PhD thesis, Chemical Engineering, Department, California Institute of Technology.
- Frank, P. (1974). *Entwurf von Regelkreisen mit vorgeschriebenem Verhalten*. G. Braun, Karlsruhe.
- García, C. (1984). Quadratic dynamic matrix control of nonlinear processes. In *Proc. AIChE Annual Meeting*.
- García, C. and Morari, M. (1982). Internal model control: 1—a unifying review and some new results. *Ind. Engng Chem. Process Res. Dev.*, **21**, 308.
- Hernández, E. (1992). Control of nonlinear systems using input–output information. PhD thesis, Chemical Engineering, Department, Georgia Institute of Technology.
- Kantor, J. (1986). Stability of state feedback transformations for nonlinear systems—some practical considerations. In *Proc. American Control Conf.*, pp. 1014–1016.
- Koh, T. and E. Powers (1985). Second-order Volterra filtering and its application to nonlinear system identification. *IEEE Trans. Acoust. Speech Sig. Process.*, **ASSP-33**, 1445–1455.
- Lo, K., Y.-K. Yeo, H.-K. Song and E. Yoon (1992). Application of a bilinear long-range predictive control method in a distillation process. In *Proc. DYCORN '92*, IFAC, pp. 363–368.
- Martin, R. (1976). *Nonlinear Operators and Differential Equations in Banach Spaces*. Wiley, New York.
- Pearson, R., B. Ogunnaike and F. Doyle III (1992). Identification of discrete convolution models for nonlinear processes. In *Proc. AIChE Annual Meeting*.
- Peterson, T., E. Hernandez, Y. Arkun and F. Schork (1992). Nonlinear dmc algorithm and its application to a semi batch polymerization reactor. *Chem. Engng Sci.*, **47**, 737–753.
- Ray, W. (1972). On the mathematical modeling of polymerization reactors. *J. Macromol. Sci.—Rev. Macromol. Chem.*, **C8**, 1–56.
- Richalet, J., A. Rault, J. Testud and J. Papon (1978). Model predictive heuristic control; application to industrial processes. *Automatica*, **14**, 413–428.
- Rugh, W. (1981). *Nonlinear System Theory—The Volterra/Wiener Approach*. The Johns Hopkins University Press, Baltimore.
- Schetzen, M. (1980). *The Volterra and Wiener Theories of Nonlinear Systems*. Wiley, New York.
- Seinfeld, J. & L. Lapidus (1974). *Mathematical Methods in Chemical Engineering*, Vol. 3: *Process Modeling, Estimation, and Identification*. Prentice-Hall, Englewood Cliffs, NJ.
- Simminger, J., E. Hernandez, Y. Arkun and F. Schork

- (1991). A constrained multivariable nonlinear model predictive controller based on iterative QDMC. In *Proc. IFAC ADCHEM Conf.*
- van de Vusse, J. (1964). Plug-flow type reactor versus tank reactor. *Chem. Engng Sci.*, **19**, 994–997.
- Wray, J., D. Sanders and G. Green (1992). Neural networks and nonlinear prediction. In *Proc. American Control Conf.*
- Yeo, Y. and C. Williams (1987). Bilinear model predictive control. *Indust. Engng Chem. Res.*, **26**, 2267–2274.
- Zafiriou, E. and A. Marchal (1991). Stability of SISO quadratic dynamic matrix control with hard output constraints. *AIChE J.* **37**, 1550–1560.