Nonparametric Nonlinear Model Predictive Control

Hiroshi KASHIWAGI* and Yun LI**
*Faculty of Engineering, Kumamoto University, 2-39-1 Kurokami, Kumamoto 860-8555, Japan,
(E-mail:kashiwa@gpo.kumamoto-u.ac.jp)
**Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow G12 8LT, UK.
(Email: Y.Li@elec.gla.ac.uk)

Keywords: Model predictive control, Process control, Nonlinear modelling, Volterra kernels, M-sequence
Abstract: Model Predictive Control (MPC) has recently found wide acceptance in industrial applications, but its potential has been much impounded by linear models due to the lack of a similarly accepted nonlinear modelling or data based technique. The authors have recently developed a new method for obtaining Volterra kernels of up to third order by use of pseudorandom M-sequence. By use of this method, nonparametric NMPC is derived in discrete-time using multi-dimensional convolution between plant data and Volterra kernel measurements. This approach is applied to an industrial polymerisation process using Volterra kernels of up to the third order. Results show that the nonparametric approach is very efficient and effective and considerably outperforms existing methods, while retaining the original data-based spirit and characteristics of linear MPC.

1 Introduction

Model Predictive Control (MPC) represents a class of control schemes where the control signal generation involves the on-line use of a parametric or a nonparametric model of the plant. Major design techniques of MPC include Model Algorithm Control, Dynamic Matrix Control, Internal Model Control and Generalised Predictive Control, etc. The underling strategy of MPC is to, at any given time, solve on-line a receding open-loop optimal control problem over a finite time horizon, where only the first control of the resulting control sequence is actually implemented on the plant. MPC algorithms are very intuitive and easy to understand, and practical constraints can often be included in the on-line open-loop algorithm[8]. Since it is straightforward to implement in industrial applications, particularly in chemical processes, where the dynamics is relatively slow and can hence accommodate on-line optimisation easily, MPC has received a worldwide attention.

However, much of the work has been confined to a linear control strategy, based on a linear model in predicting future values of the plant response[2]. Since there often exists severe nonlinearity in an industrial process that can hardly be ignored in practice, higher control performance can only be achieved through using a nonlinear model[9], including polynomial ARMA model[4], bilinear model[11], combined ARMA-Hammerstein model and neural networks. In the bid to achieve improved performance, however, these approaches sacrifice the simplicity, accuracy and characteristics arising from process I/O data. This limits the range of industrial applications, as these models are often difficult and inaccurate to obtain in practice[9] [10].

This has stimulated work on formulating MPC for use with a nonparametric Wiener model and (its more practical version) Volterra model[2] [3] [7] and on Volterra modelling [5] [6][9]. A potential advantage of using a nonparametric model is that it can yield nonlinear MPC (NMPC) directly from process I/O data. However, methods developed elsewhere require a first-principles model as so to derive a second-order Volterra model analytically from the bi-linearised fundamental model[2] [3] [7]. Hence the original limitation on the range of applications still remains. Further, these NMPC methods are so far limited to one nonlinear kernel only, i.e., up to the second-order Volterra kernels may be obtained and utilised. Hence, the fuller potential of NMPC remains yet to be unleashed.

Recently, progress on Volterra modelling with a high degree of accuracy has been made at Kashiwagi Laboratory, Kumamoto University, using which Volterra kernels of up to the third order can now be measured easily by perturbing the plant with a pseudorandom M-sequence signal that provides enough excitation and yet is acceptable in an industrial situation [6]. This progress permits Volterra NMPC schemes to be extended to the third order. In place of a Kalman estimator for the linear case, Volterra measurements offer the potential to realise NMPC the same way as linear MPC and hence the full capability of the Volterra methodology, as reported in this paper.

In the next section, objectives and solutions of NMPC are first analysed, followed by time-domain formulation using multi-dimensional convolution with Volterra kernels. For this, the M-sequence based high-order Volterra identification techniques are detailed in Section 3. Case studies are reported in Section 4 and conclusions are drawn in Section 5.

2 Nonparametric NMPC

2.1 Design Objective and Exact Solution

Refer to **Fig. 1** for notations in a general framework of model-based control. The pre-filter **F** outside the loop is for robust considerations in model-following and is often a unity-gain first-order low-pass with a relatively small time-constant or a critically damped second-order filter with a relatively high natural frequency. The process **P** is modelled by **G**. The controller **Q** generates a control sequence u through observed plant output y and output predicted by **G**. The estimator block will be interpreted in Section 2.2.

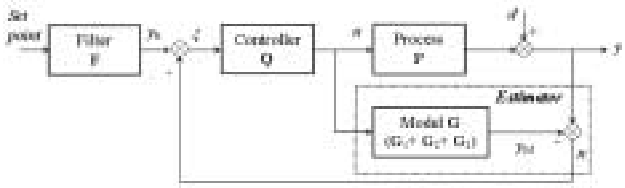

Figure 1: Model predictive control framework

Suppose that **P** has a fading memory, as indeed found in many industrial processes. Then its output can be represented by the Volterra series as a temporal extension of the Taylor series expansion, as given by

$$y(t) = \sum_{i=1}^{\infty} \int_0^{\infty} \int_0^{\infty} \cdots \int_0^{\infty} p_i(\tau_1, \tau_2, \cdots \tau_i) \times u(t-\tau_1) \cdots u(t-\tau_i) d\tau_1 \cdots d\tau_i + d(t) \quad (1)$$

where p_i is the i-th order Volterra kernel, an i-dimensional impulse response of the nonlinear process. This equation can be compactly rewritten as[2]

$$\begin{aligned} y(t) &= \sum_{i=1}^{\infty} y^{(i)}(t) + d(t) & (2) \\ &= \sum_{i=1}^{\infty} p_i * u + d(t) & (3) \end{aligned}$$

where $y^{(i)}$ is the degree-i Volterra contribution to the overall output and $*$ denotes i-dimensional convolution.

Similar to any other control schemes, the design objective, $J : R^M \to R^+$, is to find a **Q** such that

$$J = \min_{\mathbf{u}} \parallel e(t) \parallel \quad (4)$$

subject to constraints imposed upon by saturations of actuators and their change rates:

$$\begin{aligned} u_{Min} \leq \quad u \quad &\leq u_{Max} & (5) \\ \Delta u_{Min} \leq \quad \Delta u \quad &\leq \Delta u_{Max} & (6) \end{aligned}$$

where

$$\begin{aligned} u &= q * \xi & (7) \\ e &= y_R - y & (8) \end{aligned}$$

Here M is the degree of freedom in $\mathbf{u} \in R^M$ under optimisation and y_R is the desired output. Denoting the finite number of Volterra kernels of **G** that may be identified from **P** by g_i, $i = 1, ..., N$,

$$\begin{aligned} y_M &= \sum_{i=1}^{N} g_i * u & (9) \\ &= g * q * \xi \end{aligned}$$

From this and Eq.(8),

$$\begin{aligned} e &= (\xi + n) - (y_M + n) \\ &= \xi - g * q * \xi & (10) \end{aligned}$$

The objective of Eq.(4) is therefore strictly met if

$$(g * q) * \xi = \xi, \forall \xi \quad (11)$$

i.e., if an exact inverse controller $\mathbf{Q} = \mathbf{G^{-1}}$ is found.[2]

However, obtaining a strictly zero J of Eq.(4) or strictly satisfying Eq.(11) for an exact inverse would be impossible in practice, as otherwise the controller 'gain' would, under constraints, need to be infinite for all frequency and time. This is also because an exact inverse implies that the MPC will reduce to open-loop control in effect, which in turn cannot guarantee Eq.(11) or Eq.(4). Such a realisation can lead to steady-state offsets if disturbance $d(t)$ or estimation error $n(t)$ exists[8].

To resolve this problem, Doyle III *et al*[2] have decomposed **G** into degree-1 and degree-2 Volterra components, G_1 and G_2 (**Fig. 1**), and derived a 'generalised inverse' based on the 'left inverse' of the degree-1 component. Their Volterra NMPC framework developed is in a second-order analytic domain, as it needs an analytical Volterra model and this can only be identified up to the second order via the bilinear Carlemann 'linearisation' applied to a first-principles model. This means that their NMPC controller offers only one nonlinear kernel and also loses the ease of realisation present in the data-based linear MPC.

Further, this treatment still does not solve the offset and robustness problem. Unfortunately, a well-developed theory for nonlinear state observers to combat this is unavailable. While this issue remains unaddressed in the main derivations, Doyle III *et al*[2] have intuitively, in their case studies, augmented the controller with the pre-filter **F**, by moving it inside the loop to prevent an exact inverse. Clearly, this adds complexity to controller synthesis and revolts the rigour of their prior theoretical derivations. We shall show in the following section that this arrangement is unnecessary and the offset problem will be solved neatly using a nonparametric formulation.

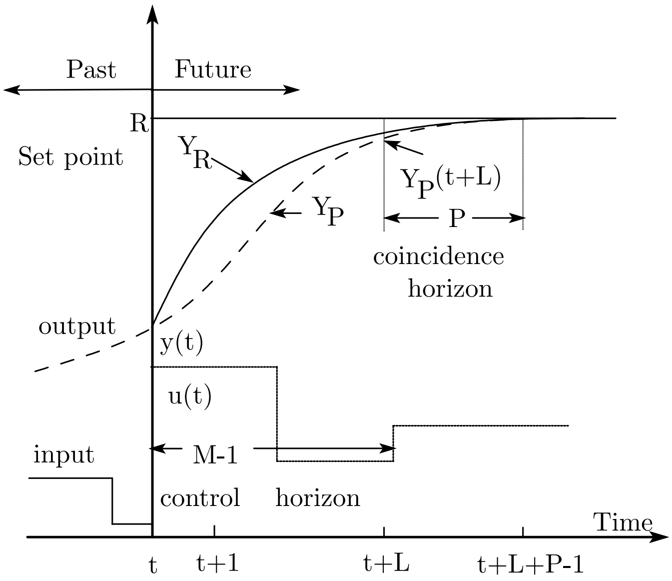

Figure 2: General principle of discrete-time MPC

2.2 Nonparametric Formulation

To relieve application engineers from needing to obtain, and linearise, a first-principle based nonlinear model, NMPC is to be formulated here using the nonparametric model given by Eq.(10). Requiring only a modest computational power in the realisation and needing not an on-line Runge-Kutta solver (as does the analytical method by Maner *et al*[7]), this should be more suitable for computer implementation and for retaining the discrete-time characteristics of linear MPC. Further, a third or higher-order Volterra controller formulated this way does not add more structural complications to a second-order one.

Discrete-time formulation of MPC can be illustrated by **Fig.2**. It requires that an 'open-loop' optimisation problem be solved on-line only over a finite prediction horizon P for a finite control horizon M. Within these receding horizons, the objective of Eq.(4) becomes

$$J_d = \min_{u(t+1|t),\ldots,u(t+M|t)} \parallel e(t+j|t) \parallel_{j=L}^{L+P-1} \tag{12}$$

where $L \geq 1$ is the minimum prediction step desired and $e(t+j|t)\ \forall j \in \{L, L+1, \cdots, L+P-1\}$ is estimated from the model and output information available at time t. Here the metric norm is evaluated within the finite discrete set $\{L, L+1, \cdots, L+P-1\}$ and may be either L_1, L_2, L_∞, mixed, weighted or any other norm provided J_d can be optimised on-line.

The open-loop offset problem encountered in an analytical inverse can be handled in discrete-time by designing an error estimator, which will give the controller an implicit integral action. As we shall see in the sequel, this is naturally and elegantly realised without needing full-state feedback, given an I/O based nonparametric, as opposed to an analytic, model. The simplest error estimator is a zero-order one, i.e., the discrepancy between the actual and modelled output at time t is used throughout the prediction horizon. Genceli and Nikolaou[3] have shown that, using such an error estimator, the closed-loop system with a Volterra controller is asymptotically stable with zero offsets if the uncertainty and its rate of change are bounded and the end-condition $y_M(\infty) = R$ is met. Here we extend this result to the generic MPC framework of **Fig. 1**.

Theorem: The linear MPC framework of Fig.1 is directly applicable to nonlinear MPC in the time-domain with a Volterra predictor

$$\begin{aligned} y(t+j|t) &= y(t) + [y_M(t+j) - y_M(t)], \\ &\quad \forall j \in \{L, L+1, \cdots, L+P-1\} \end{aligned} \tag{13}$$

and objective

$$J_d = \min_{u(t+1|t),\ldots,u(t+M|t)} \parallel \xi(t+j) - y_M(t+j) \parallel_{j=L}^{L+P-1} \tag{14}$$

The resultant nonlinear controller is asymptotically stable with zero offsets from set-point R, if the uncertainty and its rate of change are bounded and the end-condition $y_M(\infty) = R$ is met.

Proof: Eq.(13) is equivalent to

$$\begin{aligned} y(t+j|t) &\quad -y_M(t+j) = y(t) - y_M(t), \\ &\quad \forall j \in \{L, L+1, \cdots, L+P-1\} \end{aligned} \tag{15}$$

This implies that the modelling error $y(t) - y_M(t)$ estimated from the nonparametric Volterra model is held constant throughout the prediction horizon by a 'zero-order estimator'. Denote this amount of unmeasured state variables by $n(t)$ and error-correct the set-point to $y_R(t+j) - n(t) = \xi(t+j)$. This agrees with Fig.1 for the linear case. Therefore, the normed quantity in Eq.(14) evaluates to

$$\begin{aligned} y_R(t+j) &- [y(t) - y_M(t)] - y_M(t+j) \\ &= y_R(t+j) - y(t+j|t) \end{aligned} \tag{16}$$

and is hence the same as that in Eq.(12).

Note that here $y_M(t)$ can be estimated $\forall t \geq 0$ from convolution between $u(t)$ and the Volterra kernels, given the assumption that plant $\mathbf{P}$ has a fading memory and hence the Volterra model converges. While Genceli and Nikolaou have shown that prediction using a second-order Volterra model will meet the sufficient condition for robust tracking with zero offsets[3] , a third-order one will further reduce the estimation error (as we shall see in Section 3). Hence the theorem proves following their derivations[3].

The discrete-time Volterra model allows a rigorous retention of the original characteristics and spirit of linear

MPC. The Theorem signifies that the control sequence can be optimised such that y_M tracks ξ for an implicit inverse within the finite horizons, instead of mathematically formulating $y(t)$ to track $y_R(t)\ \forall t$. To further reduce the estimation error and improve robustness, the 'hard-command' of a set-point change can be replaced by a 'soft-command' trajectory y_R for the process to follow, provided $y_R(\infty) = R$.

Without loss of generality, consider a first-order low-pass pre-filter with unity-gain

$$F(s) = \frac{1}{1+\tau s} \tag{17}$$

The reference trajectory in the continuous domain is given by

$$y_R(t) = R(1 - e^{-t/\tau}) \tag{18}$$

It is not difficult to derive its discrete-time version in difference equation

$$y_R(t+1) = \alpha y_R(t) + (1-\alpha)R \tag{19}$$

where, with a given sampling interval T,

$$\alpha = 1 - \frac{T}{\tau} \tag{20}$$

Iterating the first-order equation yields

$$y_R(t+j) = \alpha^j y_R(t) + (1-\alpha^j)R, \tag{21}$$

for calculating the reference trajectory j steps ahead.

Since the control signal will go through a D/A converter, the resolution of u will be finite. Hence we can search for a discrete value of Δu for each element in $\mathbf{u} \in R^M$. A simple optimisation algorithm that accommodates constraints easily is an *a posteriori* hill-climbing algorithm, i.e., hill-climbing guided by trial-and-error. Compared with conventional nonlinear programming, *a posteriori* search takes a longer time, but is much more straightforward to implement for any objective metric under any constraints[10].

Now the only task left is to obtain Volterra kernels in Eq.(10) for $i = 1$, 2 and 3, using a pseudorandom sequence that provides the plant with enough excitation and yet are acceptable in an industrial situation[9].

3 Third-Order Volterra Model

Consider the identification of the nonlinear process of Eq.(1). In order to identify Volterra kernels $g_i(\tau_1, \tau_2, ...)$, an M-sequence is used to excite the nonlinear system with acceptable amplitude. The resultant crosscorrelation function $\phi_{uy}(\tau)$ between the input $u(t)$ and the output $y(t)$ can be written as

$$\phi_{uy}(\tau) = \overline{u(t-\tau)y(t)}$$

$$= \sum_{i=1}^{\infty} \int_0^{\infty}\int_0^{\infty}\cdots\int_0^{\infty} p_i(\tau_1, \tau_2, \cdots \tau_i) \times \overline{u(t-\tau)u(t-\tau_1)\cdots u(t-\tau_i)} d\tau_1 d\tau_2 \cdots d\tau_i \tag{22}$$

where $\bar{\ }$ denotes time average. Usually the moment of $u(t)$ is difficult to obtain, but with an M-sequence, an n-th moment of $u(t)$ yields easily. Here, the $(i+1)$th moment of the input M-sequence $u(t)$ is given by[5]

$$\overline{u(t-\tau)u(t-\tau_1)u(t-\tau_2)\cdots u(t-\tau_i)} = \begin{cases} 1 & \text{(for certain } \tau) \\ -1/N & \text{(otherwise)} \end{cases} \tag{23}$$

where N is the period of the M-sequence. For an M-sequence with the degree greater than 16, $1/N$ is in the order below 10^{-5}. Hence Eq.(23) can be approximated as a set of impulses which appear at certain τ's.

Let us consider measuring the i-th Volterra kernel. Then for integers $k_{i1}^{(j)} < k_{i2}^{(j)} < \cdots, k_{ii-1}^{(j)}$, there exists a unique $k_{ii}^{(j)} (mod\ N)$ such that[6]

$$u(t)u(t+k_{i1}^{(j)})\cdots u(t+k_{i,i-1}^{(j)}) = u(t+k_{ii}^{(j)}) \tag{24}$$

where j is the number of the group $(k_{i1}, k_{i2}, \cdots, k_{i,i-1})$ for which Eq.(24) holds. This property is called the *Shift and Add Property* of the M-sequence. Assume that the total number of those groups is m_i (that is, $j = 1, 2, \cdots, m_i$). Then Eq.(23) becomes unity when

$$\tau_1 = \tau - k_{i1}^{(j)}, \tau_2 = \tau - k_{i2}^{(j)}, \cdots \tau_i = \tau - k_{ii}^{(j)} \tag{25}$$

Therefore Eq.(22) is approximated by

$$\phi_{uy}(\tau) \simeq \sum_{i=1}^{\infty}\sum_{j=1}^{m_i} g_i(\tau - k_{i1}^{(j)}, \tau - k_{i2}^{(j)}, \cdots \tau - k_{ii}^{(j)}) \tag{26}$$

Since $g_i(\tau_1, \tau_2, \cdots \tau_i)$ is zero when any of τ_i is smaller than zero, each $g_i(\tau - k_{i1}^{(j)}, \cdots \tau - k_{ii}^{(j)})$ in Eq.(26) appears in the crosscorrelation function $\phi_{uy}(\tau)$ when $\tau > k_{ii}^{(j)}$.

In order to obtain Volterra kernels from Eq.(26), $k_{ii}^{(j)}$ must appear sufficiently apart from one another. For this to be realized, we should select suitable M-sequences that set the cross-sections of the Volterra kernels sufficiently apart from one another. Some appropriate M-sequences are given in Kashiwagi[6].

When measuring Volterra kernels up to the third order, the crosscorrelation function $\phi_{uy}(\tau)$ becomes

$$\phi_{uy}(\tau) = \Delta t g_1(\tau) + F(\tau) + 2(\Delta t)^2 \sum_{j=1}^{m_2} g_2(\tau - k_{21}^{(j)}, \tau - k_{22}^{(j)}) + 6(\Delta t)^3 \sum_{j=1}^{m_3} g_3(\tau - k_{31}^{(j)}, \tau - k_{32}^{(j)}, \tau - k_{33}^{(j)}) \tag{27}$$

where

$$F(\tau) = (\Delta t)^3 g_3(\tau,\tau,\tau) + 3(\Delta t)^3 \sum_{q=1}^{m_1} g_3(\tau,q,q) \qquad (28)$$

and Δt is the time increment or sampling period. To generalize, we have,

$$\begin{aligned} &\phi_{uy}(\tau) = \Delta t g_1(\tau) + F(\tau) \\ + &\sum_{i=2}^{\infty} i!(\Delta t)^i \sum_{j=1}^{m_i} g_i(\tau - k_{i1}^{(j)}, \cdots, \tau - k_{ii}^{(j)}) \end{aligned} \qquad (29)$$

Here $F(\tau)$ is a function of τ and is the sum of the odd order Volterra kernels when some of its arguments are equal. Since $F(\tau)$ appears together with $g_1(\tau)$ in an overlapped manner, $F(\tau)$ must be calculated from the odd order Volterra kernels and be subtracted from the measured $g_1(\tau)$ in order to obtain an accurate $g_1(\tau)$. Following this, a Volterra model can be identified for use with Eq.(13).

4 Case Study

This method is applied to a polymerization process of Mitsubishi Chemical Corp. Japan. Their chemical reactor is described by the differential equation

$$\begin{cases} \frac{dx_1}{dt} &= \frac{1}{Tp_1}(-x_1 + Kp_1 u_1) \\ \frac{dx_2}{dt} &= \frac{1}{Tp_2}(Kp_2 x_1 x_2 - x_2 + Kp_3 u_2) \\ y &= x_2 \end{cases} \qquad (30)$$

with initial condition

$$\begin{cases} x_1 = 0.02\text{kg h}^{-1}, & x_2 = 5.0\text{kg cm}^{-2} \\ u_1 = 0.05\text{kg h}^{-1}, & u_2 = 3195\text{kg h}^{-1} \end{cases} \qquad (31)$$

where x_1 is the consumption velocity of catalyst, x_2 is gas density, u_1 is the supply quantity of catalyst, u_2 is the supply quantity of polyethylene, and Tp_1, Tp_2, Kp_1, Kp_2, Kp_3 are constants. Here the control input to optimise is u_1 and the output to control is x_2, required to follow a step change in reference to $R = 10$ kg cm^{-2} from 5.

For this, an M-sequence, denoted by Δu, with amplitude ± 0.025 and characteristic polynomial $f(x) = 260577$ in octal notation is applied to the reactor, with a sampling period of 0.3 h. Taking cross-correlations between Δu and Δy, Volterra kernels are measured. These are shown in **Fig.3** $\sim$ **Fig.5**.

In **Fig.6**, comparison is made between the actual output and the Volterra estimates responding to a sinusoidal input. We see again that the third-order model offers the best estimation which should be sufficient enough to preclude the need for a further higher-order model.

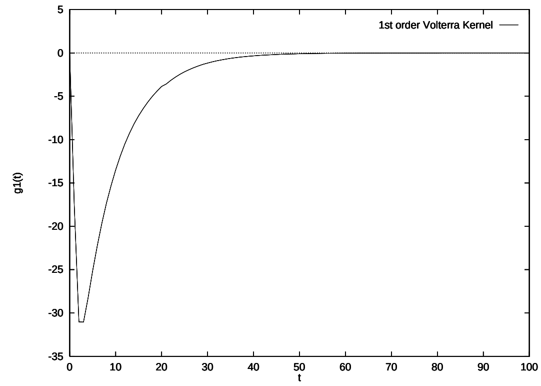

Figure 3: Obtained first-order Volterra kernel of the Mitsubishi polymerisation process

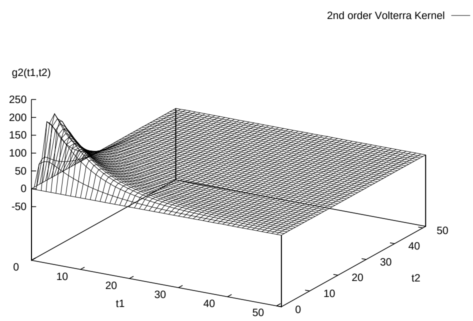

Figure 4: Obtained second-order Volterra kernel of the process

Then, using the measured Volterra kernels, NMPC is realised with the same settings in the van de Vusse case. The search for the control sequence is carried out within the range of ± 0.05 with a 0.001 increment. **Fig.7** presents the controlled performance. From the results, we see that the nonparametric NMPC formulated from the third-order Volterra model offers the best closed-loop performance.

5 Conclusion

In this paper, methods for formulating and realising nonparametric Volterra NMPC have been developed. They allow the retention of the original characteristics of linear MPC and relive practising engineers from the tedious take of obtaining an, often less accurate and less computerised, fundamental model. This simple and yet rigorous approach to NMPC incorporates an error estimator automatically in the realisation with zero offsets, needing no

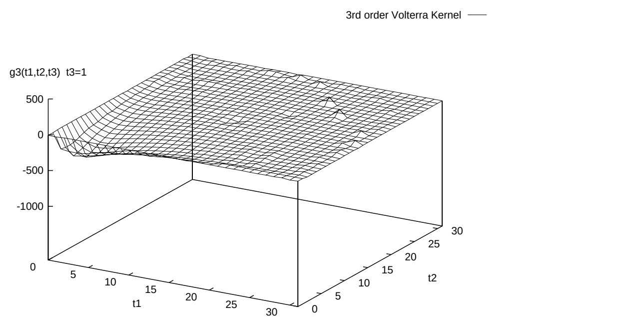

Figure 5: Obtained third-order Volterra kernel of the process

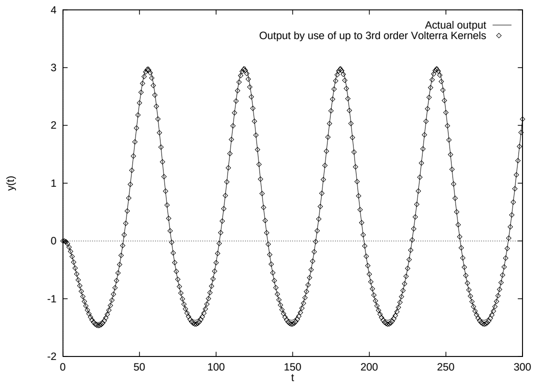

Figure 6: Comparison between the actual output and the output estimated from a third-order model

explicit nonlinear observer. The nonparametric NMPC has been applied to the control of an industrial polymerisation process using Volterra kernels of up to the third order. Results show that it is very efficient and effective and considerably outperforms existing methods. Further work includes the development of powerful nonlinear search algorithms for use with on-line implementation under multiple constraints.

References

[1] Billings, S. A., and S. Y. Fakhouri: Identification of nonlinear systems using Correlation Analysis Pseudo-random Inputs, *Int. J. Systems Sci.,* **11**-3, C261/279 (1980)

[2] Doyle III, F. J., B. A. Ogunnaike and R. K. Pearson: Nonlinear model-based control using second-order Volterra models, *Automatica,* **31**(5), 697/714 (1995)

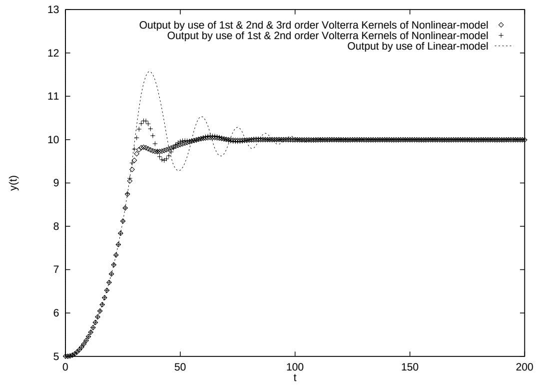

Figure 7: Performance of the NMPC controllers formulated using the first, second and third-order Volterra models for the Mitsubishi polymerisation process

[3] Genceli, H., and M. Nikolaou: Design of robust constrained model-predictive controllers with Volterra series, *AIChemE J.,* **41**-9, 2098/2107 (1995)

[4] Hernandez, E., and Y. Arkun: Control of nonlinear systems using polynomial ARMA models, *AIChE J.,* **39**, 446/460 (1993)

[5] Kashiwagi, H., and L. Rong: Identification of Volterra kernels of nonlinear van de Vusse reactor, *Trans. Contr., Automation & Syst. Eng.*, **4**-2, 109/113 (2002)

[6] Kashiwagi, H., and Y. Sun: Identification of nonlinear system by use of Volterra Kernel, *Trans. SICE*, **31**-8, 1054/1060 (1995)

[7] Maner, B. M., F. J. Doyle III, B. A. Ogunnaike and R. K. Pearson: Nonlinear Model Predictive control of a Simulated Multivariable Polymerization Reactor Using Second-order Volterra Models, *Automatica*, **32**-9, 1285/1301 (1996)

[8] Mayne, D. Q., J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert: Constrained model predictive control: stability and optimality, *Automatica*, **36**-6, 789/814 (2000)

[9] Pearson, R. K.: Selecting nonlinear model structures for computer control, *J. Process Control*, **12** (2002)

[10] Tan, K. C., and Y. Li: Grey-box model identification via evolutionary computing, *Control Engineering Practice*, **10**-7, 673/684 (2002)

[11] Yeo, Y. and C. Williams: Bilinear model predictive control, *Indust. Eng. Chem. Res.,* **26**, 2267/2274 (1987)