General Control Horizon Extension Method for Nonlinear Model Predictive Control

Hai-Tao Zhang[†,‡] **and Han-Xiong Li***[,‡]

The Key Laboratory of Image Processing and Intelligent Control, Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, People's Republic of China, and Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong

In the nonlinear model predictive control (NMPC) field, it is well-known that the multistep control approach is superior to the single-step approach when examining high-order nonlinear systems. In the multistep control approach, however, the online minimization of a 2-norm square objective function over a control horizon of length M always requires solving a set of complex polynomial equations, for which no definite solution exists so far. Moreover, the complex nature of the receding horizon optimization also causes additional problems to its closed-loop stability analysis. With these two serious challenges in mind, using a Volterra−Laguerre model-based NMPC for discussion, we propose a general technique to extend the control horizon with the assistance of Groebner basis, which transforms the set of complex polynomial equations to a much simpler form. We prove the closed-loop stability of the algorithm in the sense that the input and output series are both mean-square-bounded. Finally, the efficiency of this improved algorithm is examined on an industrial constant-pressure water supply system. Compared to the conventional NMPC schemes, the proposed method with the control horizon extension has shown a great potential to control a wide range of nonlinear dynamic systems.

1. Introduction

Model predictive control (MPC) is an attractive optimization strategy, particularly for nonlinear processes in real-world applications. Conventional MPC schemes have enjoyed widespread acceptance and success as an effective technique for addressing control problems, especially in the chemical/petrochemical industries.[1] Recently, research has focused on the nonlinear MPC (NMPC) problem; however, the derivation of these models can be very time-consuming, especially if the process is not well-understood. Furthermore, the NMPC schemes that use more-realistic nonlinear process descriptions always sacrifice the simplicity associated with linear techniques, to achieve the improved performance.[2] The direct use of nonlinear models often leads to a high-order nonlinear optimization problem, which has not been easy to solve so far.[2]

To enhance the efficiency of NMPC, more and more empirical nonlinear models are used,[3] such as the nonlinear auto-regressive moving average with exogenous inputs (NARMAX) model,[3] the nonlinear auto-regressive with exogenous inputs (NARX) model,[3] the Volterra series model,[3,4] and the Wiener model,[3,4] among others. Because these discrete-time models are of high-order, a multistep input series (a multistep control horizon) rather than a single-step input is required to predict system output over certain future steps. The extension of receding techniques to the multistep control situation will be very helpful to handle high-order nonlinearities. However, this extension is not easy to achieve, for the following two reasons.

(1) The online optimization of the object function always leads to a set of complex polynomial equations, including coupling of the inputs at different steps,[1,2,7] for which no definite solution is available so far.

(2) Even if this control law is available, another serious challenge still remains on the closed-loop stability analysis, which involves the receding horizon optimization nature of NMPC.[8]

To meet these two challenges, one of the most frequently studied cases—the Volterra series model-based NMPC algorithm[6]—will be investigated as an example. The Volterra series model[4,5] is a nonlinear extension of the linear impulse response model, which is capable of capturing the nonlinear dynamics with fading memory[4] (i.e., the effects of past inputs on the output are negligible after some finite time). In our study, each Volterra kernel is expanded by a complete orthonormal series of functions called the Laguerre functional series,[9−11] which is chosen based on the effective synthesis of causal operators. Although many orthornormal sets, such as Kautz series[10,11] and Tschebyscheff series,[9] are also available, a proper Laguerre filter pole will result in fewer modeling parameters,[12,13] if the higher-order Volterra kernels can be neglected.

During the past few years, Laguerre filters have been successfully applied to design linear adaptive controllers,[14−19] which has caused interest of modeling stable linear plants using Laguerre filters.[20−22] Compared with the FIR (finite impulse) or the ARMA (autoregressive moving average) model, the Laguerre model is good at approximating systems with varying time delay with the following advantages:[6,14,15,17] (i) tolerance to unmodeled dynamics and reduced sensitivity of the estimated parameters, (ii) orthogonality of the regression vector under white excitation, (iii) no assumption is required for model order and process delay, and (iv) good low frequency match exists between the estimated model and the real plant.

* To whom correspondence should be addressed. Tel.: 852-27888435. Fax: 852- 27888423. E-mail address: mehxli@cityu.edu.hk.

[†] The Key Laboratory of Image Processing and Intelligent Control, Department of Control Science and Engineering, Huazhong University of Science and Technology.

[‡] Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong.

10.1021/ie0703944 CCC: $37.00 © 2007 American Chemical Society
Published on Web 11/10/2007

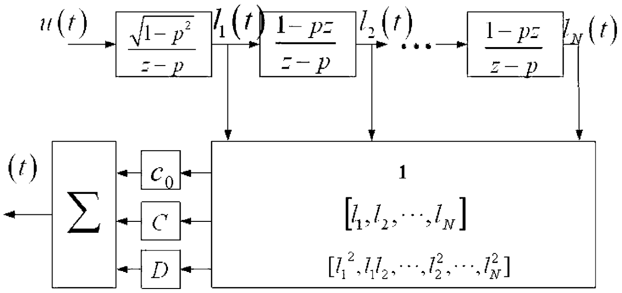
u(t)
$\frac{\sqrt{1-p^2}}{z-p}$
$l_1(t)$
$\frac{1-pz}{z-p}$
$l_2(t)$
$\frac{1-pz}{z-p}$
$l_N(t)$
$y_m(t)$
Σ
c_0
C
D
1
$[l_1, l_2, \cdots, l_N]$
$[l_1^2, l_1 l_2, \cdots, l_2^2, \cdots, l_N^2]$

Figure 1. Volterra−Laguerre nonlinear model in the discrete-time domain.

The Volterra−Laguerre model was first proposed and analyzed by Schetzen[5] in 1980. Boyd and Chua[3,4] proved, in 1984, its superiority, in comparison with other empirical models, such as the NARMAX model and the NARX model, when capturing the dynamics of fading memory nonlinear systems (FMNSs), which are defined as systems whose dependence on past inputs decreases rapidly enough with time.[3] Thereafter, more and more researchers recognized the potential of the Volterra−Laguerre model in nonlinear process modeling and control. The most typical example is Dumont's NMPC[6] with the single-step control horizon, which was successfully applied to a wood chip refiner motor load control system for mechanical pulping. However, because of the complexity of receding horizon optimization nature, the two general theoretical issues of NMPC, namely (a) control horizon extension (and, in particular, finding the solution of complex polynomials of equations) and (b) closed-loop stability analysis still remain unsolved,[6,7,23−26] which greatly limits its further application.

Bearing these problems in mind, this paper presents a general method to extend the control horizon of NMPC algorithms with guaranteed stability. The main contributions of this paper are summarized as follows:

(1) The traditional single-step control NMPC algorithm is improved and changed to a multistep one. This is done using an important concept from the algebraic geometric—namely, a Groebner basis[27,28]—which allows the mentioned sets of complex polynomial equations to be transformed to simpler forms with easy solution.

(2) Based on a lemma proposed by Goodwin,[29,30] the stability of this closed-loop system is proved in the sense that the input and output series are both mean-square-bounded.

Hence, this novel method can be expected to take full advantage of NMPC technology and to be capable of examining more-complex nonlinear dynamics. The developed algorithm will be demonstrated with promising results on a real water supply system.

The remainder of the paper is organized as follows. In section 2, the Volterra−Laguerre Model is first introduced. In section 3, two main problems are described, together with the conventional single-step control NMPC algorithm. In section 4, the improved multistep control NMPC algorithm then is proposed, together with the theoretical analysis of the closed-loop stability. Experiments are performed in section 5. Finally, conclusions are given in section 6.

2. Volterra−Laguerre Modeling for Processes

If the dependencies of the process dynamics on past inputs decrease rapidly enough with time, the input−output relationship of this nonlinear process can be approximated by Volterra series[2] as

$$y_m(t) = h_0(t) + \sum_{n=1}^{\infty} \int \cdots \int h_n(\tau_1,...,\tau_n) \prod_{i=1}^{n} u(t-\tau_i)\, \mathrm{d}\tau_i \quad (1)$$

where the functions $h_n(\tau_1,...,\tau_n)$ are the Volterra kernels that represent the nonlinear dynamics. This type of system is called FMNS,[2] which is well-behaved in the sense that it will not exhibit multiple steady states or other related phenomena, such as chaotic responses. Fortunately, most industrial processes (pH neutralization processes, continuously stirred tank reactor (CSTR) processes, distillation processes, heat exchange processes, etc.) belong to FMNS systems.[20] In practice, a Volterra series should be truncated at a finite value N_v.[1]

We denote the ith-order Laguerre time function by $\phi_i(t)$ and the ith-order Laguerre filter, which is shown in the upper part of Figure 1, by

$$l_i(t) = \int_0^{\infty} \phi_i(\tau) u(t-\tau)\, \mathrm{d}\tau \quad (2)$$

where the expression of $\phi_i(t)$ is given in refs 9 and 10. Because $\{\phi_i\}$ forms a complete orthonormal set in the space $\mathbf{L}_2(\mathbb{R}^+)$, we can write, under the assumption that the Volterra kernels[5] are stable,

$$\begin{aligned} h_1(\tau_1) &= \sum_{i=1}^{N} c_i \phi_i(\tau_1) \\ h_2(\tau_1,\tau_2) &= \sum_{n=1}^{N} \sum_{k=1}^{N} c_{nk} \phi_n(\tau_1) \phi_k(\tau_2) \\ &\vdots \end{aligned} \quad (3)$$

where c_i, c_{nk}, ... are constant coefficients.[5] The expansion error goes to zero as N and N_v approach infinity; here, N is defined as the Laguerre series truncation length. Using the orthonormal property of the Laguerre function, and substituting eqs 2 and 3 into eq 1, the input−output model becomes[5]

$$y_m(t) = c_0(t) + \sum_{i=1}^{N} c_i l_i(t) + \sum_{n=1}^{N} \sum_{k=1}^{N} c_{nm} l_n(t) l_k(t) + \cdots \quad (4)$$

Because of the fact that Volterra kernels are symmetric, we define

$$\begin{aligned} L(t) &= [l_1(t),...,l_N(t)]^{\mathrm{T}} \\ C &= [c_1,...,c_N]^{\mathrm{T}} \\ \mathbf{D} &= \begin{bmatrix} c_{11} & \cdots & c_{1N} \\ & \ddots & \vdots \\ * & & c_{NN} \end{bmatrix} \end{aligned} \quad (5)$$

Here, the symbol '*' denotes "transpose" and **D** is a symmetric matrix.

Assume the truncation length of the Volterra series is $N_v = 2$; the Volterra−Laguerre model then becomes[6,24,25]

$$\dot{L}(t) = \mathbf{A_c} L(t) + \mathbf{B_c} u(t) \quad (6)$$

$$y_{\mathrm{m}}(t) = c_{c0} + \mathbf{C}_{\mathrm{c}}^{\mathrm{T}} L(t) + L^{\mathrm{T}}(t) \mathbf{D}_{\mathrm{c}} L(t) \quad (7)$$

Based on this, we similarly consider the discrete-time version[11,17] of eqs 6 and 7:

$$L(t + 1) = \mathbf{A}L(t) + \mathbf{B}u(t) \tag{8}$$

$$y_m(t) = c_0 + \mathbf{C}^{\mathrm{T}}L(t) + \mathbf{L}^{\mathrm{T}}(t)\mathbf{D}L(t) \tag{9}$$

with

$$\mathbf{A} = \begin{bmatrix} p & 0 & 0 & \cdots & 0 \\ \beta & p & 0 & \cdots & 0 \\ -p\beta & \beta & p & \cdots & 0 \\ p^2\beta & -p\beta & \beta & \cdots & 0 \\ \vdots & & & & \\ (-1)^{N\ -2}p^{N-2}\beta & (-1)^{N-3}p^{N-3}\beta & \cdots & \beta & p \end{bmatrix} \tag{10a}$$

$$\mathbf{B} = \begin{bmatrix} \beta^{1/2} \\ (-p)\beta^{1/2} \\ \vdots \\ (-p)^{N-1}\beta^{1/2} \end{bmatrix} \tag{10b}$$

$$\beta = \sqrt{1 - p^2} \tag{10c}$$

where p is the Laguerre filter pole (see the upper portion of Figure 1). The detailed model structure of Figure 1 can be explained as follows: the upper portion (the Lagurrre filters) calculates the state vector $L(t)$; accordingly, the zero-order kernel (1), the first-order kernel ($[l_1(t),\cdots,l_N(t)]$), and the second-order kernel $[l_1{}^2(t),l_1(t),l_2(t),..., l_2{}^2(t),..., l_N{}^2(t)]$ are computed within the lower portion of Figure 1. Finally, combining these kernels and their corresponding coefficients c_0,$\mathbf{C}$ and $\mathbf{D}$ yields the model output $y_m(t)$. Generally, the initial values $L(0)$ can be pre-optimized as[17]

$$L(0) = \sqrt{1 - p^2}\ [1, -p, p^2, ..., (-1)^{N-1}p^{N-1}]^{\mathrm{T}}$$

Here, a finite number of Laguerre filters are used, indicating that the true plant is stable and observable in finite time. Equations 8−10 are an approximation in input−output form of the Volterra functional series representation for a nonlinear dynamic system.

Fortunately, the model parameters c_0, $\mathbf{C}$, and $\mathbf{D}$ are in a linear regressive form, which can be easily estimated by least-squares estimation (LSE)[31] as follows:

$$y(t) = \theta^{\mathrm{T}}\Phi(t) \tag{11}$$

with

$$\theta = [c_0,c_1,\cdots,c_N,c_{11},\cdots,c_{1N},c_{21},\cdots,c_{2N},\cdots,c_{NN}] \tag{12}$$

$$\Phi(t) = [1,l_1(t),\cdots,l_N(t),l_1{}^2(t),l_1(t)l_2(t),\cdots, l_1(t)l_N(t),l_2(t)l_1(t),\cdots,l_2(t)l_N(t),\cdots,l_N{}^2(t)] \tag{13}$$

According to eq 8, $\Phi(t)$ can be calculated with $u(t)$ at each sampling period, and the coefficients θ can be identified by recursive LSE with a forgetting factor λ.

Remark 1: Each stable Volterra kernel in the space $\mathbf{L}_2(\mathbb{R}^+)$ can be accurately approximated by a more-general type of model called an orthonormal functional series (OFS).[9−11] Pulse series, Laguerre series, and Kautz series are three typical OFSs, with orders of 0, 1, and 2, respectively. In addition, Heuberger et al.[5] proposed a method to generate high-order OFSs. With the increase in order, the OFS model can handle more-complex dynamics.

3. Problem Description

Generally speaking, to predict the system output $y(t)$, an input series rather than a single-step input is required in MPC. Thus, it seems obvious that the multistep control method is superior to the single-step approach.

First, a M-step control and a P-step prediction NMPC are designed, based on the Volterra−Laguerre model (eqs 8 and 9). The definitions of control and prediction horizons and the general derivation procedure of MPC can be reviewed in Camacho and Bordons.[1] The future state $L(t + i|t)(i = 1, ..., P)$ in eq 14 can be predicted by combining the current state $L(t)$ with the past control inputs $u(t - 1)$ and the future changes $\{\Delta u(t + j|t),j = 0, ..., M - 1\}$,

$$L(t + i|t) = \mathbf{A}^iL(t) + \bar{\mathbf{A}}_{\mathbf{i}}\mathbf{B}u(t - 1) + \sum_{j=0}^{i-1}\bar{\mathbf{A}}_{i-j}\mathbf{B}\Delta u(t + j|t) \tag{14}$$

assuming that the input changes $\Delta u(t + j|t) = 0$ for $j \geq M$, and $\bar{\mathbf{A}}_{\mathbf{i}} = \sum_{j=0}^{i-1}\mathbf{A}^j$.

If $N_{\mathrm{v}} = 2$, the model prediction (eq 14) then can be rewritten as

$$\mathbf{L}(t + 1|t) = \begin{bmatrix} \mathbf{A} & \bar{\mathbf{A}}_1\mathbf{B} \\ \mathbf{A}^2 & \mathbf{A}_2\mathbf{B} \\ \vdots & \vdots \\ \mathbf{A}^P & \bar{\mathbf{A}}_{P-1}\mathbf{B} \end{bmatrix}_{PN\times(N+1)} \cdot [L^{\mathrm{T}}(t)\ \ u(t - 1)\ ]^{\mathrm{T}} + \begin{bmatrix} \bar{\mathbf{A}}_1\mathbf{B} & 0 & \cdots & 0 \\ \bar{\mathbf{A}}_2\mathbf{B} & \bar{\mathbf{A}}_1\mathbf{B} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ \bar{\mathbf{A}}_M\mathbf{B} & \bar{\mathbf{A}}_{M-1}\mathbf{B} & \cdots & \bar{\mathbf{A}}_1\mathbf{B} \\ \vdots & \vdots & & \vdots \\ \bar{\mathbf{A}}_P\mathbf{B} & \bar{\mathbf{A}}_{P-1}\mathbf{B} & \cdots & \bar{\mathbf{A}}_{P-M+1}\mathbf{B} \end{bmatrix}_{PN\times M} \Delta\mathrm{U}(t|t) \tag{15}$$

with

$$\mathbf{Y}_{\mathrm{m}}(t + 1|t) = \mathbf{C_P}^{\mathrm{T}}\mathbf{L}(t + 1|t) + \mathbf{L}^{\mathrm{T}}(t + 1|t)\mathbf{D_P}\mathbf{L}(t + 1|t) \tag{16a}$$

$$\mathbf{L}(t + 1|t) = [L^{\mathrm{T}}(t + 1|t),...,L^{\mathrm{T}}(t + P|t)]^{\mathrm{T}} \tag{16b}$$

$$\Delta U(t|t) = [\Delta u(t|t),...,\Delta u(t + M - 1|t)]^{\mathrm{T}} \tag{16c}$$

$$\mathbf{Y}_m(t + 1|t) = [y_m(t + 1|t),...,y_m(t + P|t)]^{\mathrm{T}} \tag{16d}$$

$$\mathbf{C_P}^{\mathrm{T}} = \mathrm{diag}(\mathbf{C^T},...,\mathbf{C^T})_{P\times PN} \tag{16e}$$

$$\mathbf{D_P} = \mathrm{diag}(\mathbf{D},...,\mathbf{D})_{PN\times PN} \tag{16f}$$

The associated quadratic objective function is

$$\min_{\Delta U(t|t)} J(\Delta U(t|t)) = ||\mathbf{Y}_{\mathrm{r}}(t + 1) - \hat{\mathbf{Y}}_{\mathrm{m}}(t + 1|t)||_{\mathrm{Q}}{}^2 + ||\Delta U(t|t)||_{\mathrm{R}}{}^2 \tag{17}$$

where the rectified model output prediction vector is $\hat{\mathbf{Y}}_{\mathrm{m}}(t + 1)$

$= \mathbf{Y}_m(t+1) + e(t)$, with modeling error $e(t) = y(t) - y_m(t)$, $\mathbf{\Omega} = [1, ..., 1]^T_{P\times 1}$; the softening reference vector is

$$\mathbf{Y}_r(t+1) \triangleq [y_r(t+1), ..., y_r(t+P)]^T$$

with $y_r(t+i) = \alpha^i y(t) + (1-\alpha^i)\chi(t)$ (for $i = 1, ..., P$), α is the softening parameter, $\chi(t)$ is the set point, and $\mathbf{Q}$ and $\mathbf{R}$ are both positive-definite symmetric weighting matrices, which are always set as $\mathbf{Q} = \mathbf{I}$, $\mathbf{R} = \mathbf{r} \cdot \mathbf{I}$ for convenience. The first and second terms of eq 17 intend to penalize the error between the set point and output and penalize the control efforts, respectively.

The $\Delta U(t|t)$ minimizing the objective function (eq 17) can be obtained by solving $\partial J/\partial \Delta U(t|t) = 0$. Because, at each step, J in eq 17 is a quadratic polynomial of the control variables $\Delta u(t|t), ..., \Delta u(t+M-1|t)$, it is obvious that $\partial J/\partial \Delta U(t|t) = 0$ leads to a set of M cubic equations

polynomial equation 1:

$$f_1 = \sum_{i_1=0}^{3} \cdots \sum_{i_M=0}^{3} s^{(1)}_{i_1 i_2 \cdots i_M} \Delta u^{i_1}(t|t) \Delta u^{i_2}(t+1|t) \cdots \Delta u^{i_M}(t+M-1|t) = 0$$

$$\vdots \quad \vdots$$

polynomial equation M:

$$f_M = \sum_{i_1=0}^{3} \cdots \sum_{i_M=0}^{3} s^{(M)}_{i_1 i_2 \cdots i_M} \Delta u^{i_1}(t|t) \Delta u^{i_2}(t+1|t) \cdots \Delta u^{i_M}(t+M-1|t) = 0 \quad (18)$$

with $0 \le i_1 + i_2 + \cdots + i_M \le 3, i_j \in \mathrm{N}^+ \cup \{0\} (j = 1, ..., M)$, and the rational coefficients $s^{(j)}_{i_1, i_2, \cdots, i_M} (j = 1, ..., M)$ are determined by $u(t-1)$, $L(t)$, $e(t)$, r and $\mathbf{Y}_r(t+1)$. Unfortunately, there is so far no definite solution for the polynomial set (18), and the underlying difficulty lies on the crossing terms of different steps' control signals.

In summary, there are two unsolved problems related to the NPMC:

(1) Problem 1: Solution of Polynomial Equations. To do so, obtain the multistep control law $\Delta u(t|t)$ from eq 18.

(2) Problem 2: Stability of the control law. Prove the closed-loop stability of the above control law.

Remark 2:

(1) Actually, problems 1 and 2 are the common problems for most of the empirical model-based NMPCs.

(2) If $M = 1$, then problem 1 becomes a simpler single-step control:

$$\frac{\partial J}{\partial \Delta u(t|t)} = s_3(t)\Delta u^3(t|t) + s_2(t)\Delta u^2(t|t) + s_1(t)\Delta u(t|t) + s_0(t) \quad (19)$$

The solution and application of this case study are presented in detail in the work of Dumont and Fu[6] and Parker and Doyle.[25] However, this single-step control method cannot handle high-order nonlinear dynamics, because the control horizon is only one step. Moreover, because of the complex nature of receding horizon optimization, the closed-loop stability analysis of this algorithm has not been given yet.

4. Controller Development and Theoretical Analysis

4.1. Multistep Control NMPC Using a Groebner Basis (Problem 1). An important concept from algebraic geometry, called the Groebner basis, can be applied to solve the complex set of eq 18 in a straightforward manner. The detailed description of the Groebner basis and the relevant contents are given in Appendix A and refs 27 and 28. Actually, $f_i(1 \le i \le M)$ in eq 18 forms an ideal on a polynomial ring, whose definition is also shown in Appendix A. The Buchberger algorithm[27] then could be applied to generate the Groebner bases, which transforms the complex equation set (eq 18) into simpler basis polynomials that represent the solutions for the manipulated variable profile.

In this scheme, the lexicographic order (which is a type of relative importance order; see Appendix A) of $\{\Delta u(t|t), ..., \Delta u(t+M-1|t)\}$ is generally set as $\Delta u(t+M-1|t)$ "is more important than" $\Delta u(t+M-2|t)$ "is more important than" ... "is more important than" $\Delta u(t|t)$. A set of M Groebner basis polynomials, including two sets of equations, then is generated as follows:

(1) An ηth-order polynomial equation in only $\Delta u(t|t)$, i.e.,

$$\sum_{i=0}^{\eta} \alpha_i \Delta u^i(t|t) = 0 \qquad (\text{for } \alpha_\eta \ne 0) \qquad (20)$$

(2) $(M-1)$ polynomials equations in

$$\Delta u(t+1|t), \cdots, \Delta u(t+M-1|t) \qquad (21)$$

where $\alpha_i (i = 0, 1, ..., \eta)$ are determined by the coefficients $s^{(1)}_{i_1, i_2, \cdots, i_M}, ..., s^{(M)}_{i_1, i_2, \cdots, i_M}$ of eq 18.

Thus, eq 20 can be used to get $\Delta u(t|t)$, which would lead to its corresponding prediction $(M-1)$ steps prediction of input series, i.e., $\{\Delta u(t+1|t), ..., \Delta u(t+M-1|t)\}$, according to eq 21. To ensure the existence of a real root, M should be adjusted to make η an odd number. In this way, the structure of Groebner bases is formed to provide the straightforward solution of problem 1. In practical applications, one should not be intimidated by the complex forms of eq 18 and its corresponding Groebner basis (eqs 20 and 21), because this solution can be easily implemented with the existing software packages (e.g., *GroebnerBasis ( )* in *Mathematica*). Moreover, because eq 20 is dependent only on $\Delta u(t|t)$, the $\Delta u(t|t)$ can be obtained using a function such as *roots( )* in MATLAB. The only real roots of eq 20 are evaluated and substituted into eq 21. These equations calculate candidates $\Delta u(t+1|t), ..., \Delta u(t+M$ - $1|t)$ for each $\Delta u(t|t)$. Finally, only $\Delta u(t|t)$ that produces the minimum object function is implemented in the system. This process is repeated at each sampling period. The detailed procedure of online calculating is shown in Figure 2: first, $\partial J/\partial \Delta U(t|t) = 0$ is used to compute the M cubic equations (eqs 18), and then the Groebner basis technique is applied to get the polynomial described in eq 20; finally, eq 20 is solved to obtain the control law $\Delta u(t|t)$.

On the other hand, input magnitude constraints can be also included in the current algorithm. The optimal solution to the constrained problem is equivalent to finding the global minimum within the constrained region. If the minimum does not lie within the constrained region, then the objective function is evaluated at each constraint, and the manipulated variable move corresponding to the minimum objective function is implemented. Compared to some widespread nonlinear programming-based MPC algorithms using the objective function (eq 17), the advantage of the current method is obvious, because gradient descent-based nonlinear programming solvers can only converge to a local minimum, whereas the current analytic solution controller guarantees convergence to the global minimum objective function value. More details can be seen in the work by Parker.[7]

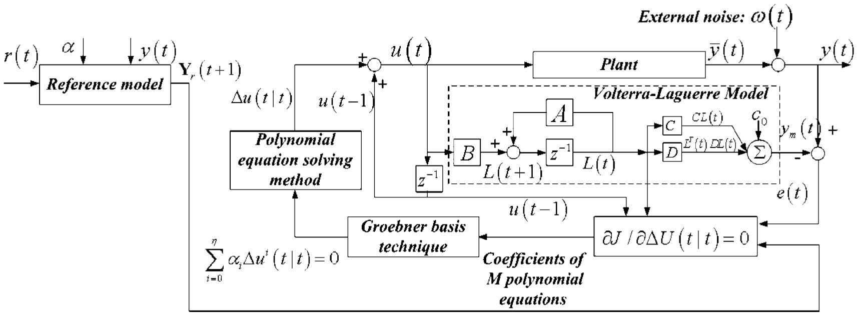

Figure 2. Multistep control algorithm structure.

Remark 3:

(1) In contrast to Parker,[7] the advantages of this work are given as follows: (a) The control horizon is extended to multisteps, rather than just two steps ahead, which makes the method more flexible. Moreover, this method can be even generalized to more-general open-loop stable NARMAX model-based NMPCs.[1−3] (b) A theorem is proposed to guarantee the closed-loop stability of this method, and parameters can be adjusted according to this theorem.

(2) The multivariable problem is a straightforward extension of the current formulation, although analytic solution of the vector objective function (eq 17) requires further work.

4.2. Closed-Loop Stability Analysis (Problem 2). For the proposed multistep control algorithm (eq 20), the theoretical issues regarding problem 2 will be discussed here.

Lemma 1.[29,30] For a discrete-time system

$$T(z^{-1})\Delta u^{\mu}(t) = H(z^{-1})y(t + P_1) + \sum_{i=1}^{\mu-1} T_i(z^{-1})\Delta u^i(t) + D(z^{-1})\omega(t) \quad (22)$$

where μ and P_1 are positive integers; $\{\Delta u(t)\}$ and $\{y(t)\}$ are incremental input series and output series of the system, respectively; $H(z^{-1})$, $D(z^{-1})$, and $T_i(z^{-1})$ (for $i = 1, ..., \mu$) are polynomials in z^{-1}; $\{\omega(t)\}$ is stochastic white noise defined in the probability space $(\Omega,\mathbb{F},\boldsymbol{P})$; $\{\mathbb{F}_t\}$ is the sub σ-algebra series of $\mathbb{F}$; and $\mathbb{F}_t$ contains all information of this system up to time t. In addition, $\{\omega(t)\}$ satisfies the following three assumptions:

$$\text{(A1)}\quad \mathbf{E}\{\omega(t)/\mathbb{F}_{t-1}\} \overset{\text{as}}{=} 0 \ (\mathbf{E}\{\cdot\} \text{ denotes the expectation value}) \quad (23)$$

$$\text{(A2)}\quad \mathbf{E}\{\omega^2(t)/\mathbb{F}_{t-1}\} \overset{\text{as}}{=} \rho^2 (0 < \rho < \infty) \quad (24)$$

$$\text{(A3)}\quad \lim_{N\to\infty} \sup \frac{1}{N}\sum_{t=1}^{N} \omega^2(t) \overset{\text{as}}{<} \infty \quad (25)$$

where the superscript “as” denotes “asymptotically”. If all the zeros of $T(z^{-1})$ are inside the unit circle of the complex plane of the Z-domain, then

$$\frac{1}{N}\sum_{k=1}^{N} (\Delta u^{\mu}(t))^2 \overset{\text{as}}{\leq} \frac{K_1}{N}\sum_{k=1}^{N} y^2(t + d) + K_2 \quad (26)$$

with $0 < K_1 < \infty$ and $0 < K_2 < \infty$.

■

Based on the Volterra−Laguerre model (eqs 8−10), it can be obtained that the closed-loop system determined by the algorithm (eq 19 or 20) has a similar form of eq 22, as shown below (a detailed derivation is given in Appendix B):

$$T(z^{-1})\Delta u^{\mu}(t) = H_0(z^{-1})y_{\mathrm{r}}(t + P) + \sum_{i=1}^{\mu-1} T_i(z^{-1})\Delta u^i(t) + D_0(z^{-1})\omega(t) \quad (27)$$

For the traditional single-step algorithm (eq 18), $\mu = 2$; for our multistep control law (eq 20), $\mu = \eta$. Note that assumptions A1−A3 determine the stochastic characteristics of the external noise, whereas Lemma 1 will no longer be valid for colored or unbounded noise.

According to the general form (eq 27) of the closed-loop system, combined with Lemma 1, the stability theorem of our algorithm can be given as follows.

Theorem 1. If the closed-loop system determined by the control law (eq 19 or 20) satisfies assumptions A4−A6, then it can be concluded that the system is closed-loop stable in the sense that the input series $\{u^i(t)\}(i = 1, ..., \mu)$ and output series $\{y(t)\}$ are both mean-square bounded.

(A4) the external stochastic white noise $\omega(t)$ satisfies assumptions A1, A2, and A3, and the modeling error $e(t)$ solely contains $\omega(t)$, i.e.,

$$y(t) - y_{\mathrm{m}}(t) = \omega(t) \quad (28)$$

(A5) the initial control law $u(0)$ is bounded, and

(A6) there exist parameters M, P, N, r, p such that all of the zeros of the polynomial $T(z^{-1})$ in eq 27 are inside the unit circle of the Z-domain complex plane.

Proof: From Lemma 1, considering that the reference $y_r(t + i)$, $(i = 1,\cdots,P)$ are bounded, we have that the series $\{\Delta u^i(t)\}(i = 1, ..., \mu)$ is mean-square bounded if assumption A6 is satisfied. Furthermore, from assumption A5, it can be obtained that $\{u^i(t)\},(i = 1, ..., \mu)$ are also mean-square bounded. On the other hand, in accordance with assumption A4, and taking into consideration of Appendix B, we have

$$y(t) = c_0 + \mathbf{C}^{\mathrm{T}} \sum_{i=0}^{N_\epsilon} \mathbf{A}^{\mathbf{i}} z^{-i-1}\mathbf{B}u(t) + \left(\sum_{i=0}^{N_\epsilon} \mathbf{A}^{\mathbf{i}} z^{-i-1}\mathbf{B}\right)^{\mathrm{T}} \mathrm{D}\left(\sum_{i=0}^{N_\epsilon} \mathbf{A}^{\mathbf{i}} z^{-i-1}\mathbf{B}\right) u^2(t) + \omega(t) \quad (29)$$

where N_ϵ is a sufficiently large positive integer (see Appendix B for details). Then, from assumption A4, combined with the fact that both N_ϵ and N_v are finite positive integers, it can be concluded that the output series {y(t)} are also mean-square bounded.

■

Remark 4: The control law (eq 27) is a closed-loop controller, because the modeling error $e(t) = y(t) - y_m(t)$ is fed back to compute the control law. Moreover, although $\omega(t)$ is directly related to $y(t)$, the conditions of Theorem 1 can still be satisfied. Let us explain as follows.

As shown in Figure 2, the system output is expressed as

$$y(t) = \bar{y}(t) + \omega(t)$$

where $\bar{y}(t)$ is the nominal output and $\omega(t)$ is the external noise. On the other hand, generally speaking, the modeling error $e(t)$ contains unmodeled dynamics $\xi(t)$ and external noise $\omega(t)$, i.e.,

$$e(t) = \xi(t) + \omega(t)$$

Here, $\xi(t)$ denotes the difference between $\bar{y}(t)$ and model output $y_m(t)$, i.e.,

$$\xi(t) = \bar{y}(t) - y_m(t)$$

Actually, in Theorem 1, it is assumed that the Volterra−Lageurre model matches the nominal model of the plant, i.e.,

$$\xi(t) = \bar{y}(t) - y_m(t) = 0$$

Thus, $e(t) = \omega(t)$. This assumption is reasonable, because the Laguerre−Volterra model has been proved to have the capability of accurately approximating arbitrary FMNSs with sufficiently large N and N_v (see refs 4 and 5). In industrial engineering applications, this assumption can be interpreted as "in contrast to $||\omega(t)||$, $||\xi(t)||$ is negligible". Consequently, external noise $\omega(t)$ is directly related to $y(t)$ as

$$y(t) = \bar{y}(t) + \omega(t)$$

Meanwhile, the conditions of Theorem 1 can be satisfied.

Remark 5: Let us give the interpretations of assumptions A4−A6. With the assistance of assumption A4, Problem 2 is naturally transformed to the framework of Lemma 1. However, if the modeling error $e(t)$ contains components other than external white noise $\omega(t)$, the problem will become more complicated. Because $\Delta u(t)$ is proven to be bounded at each step, assumption A5 guarantees the mean-square boundedness of $u(t)$. Moreover, according to Lemma 1, assumption A6 ensures the mean-square boundedness of $y(t)$.

In real applications, to decrease the computational complexity, one can first set the parameters N, M, and P as integers in some limited ranges, such as [5,10], [2,15], and [M,15], respectively, and then set the value of p around the optimal value gained by Campello and co-workers.[12,13] Afterwards, tune parameter r until one finds a suitable parameter combination of (p,N,M,P,r) that satisfies assumption A6. Note that, in the process of adjusting r, one can still slightly tune p around the optimal value to accelerate the parameter setting procedure. In contrast to the traditional single-step algorithm (eq 19), the control horizon parameter M of eq 20 can be designed to enlarge the stable region according to Theorem 1. Consequently, our improved algorithms (eq 20) have the potential to handle more-complex nonlinear dynamics, which will also be validated by the case studies in section 5.

5. Case Study

5.1. Constant-Pressure Water Supply Control System. The two NMPC algorithms to be examined are stated as follows:

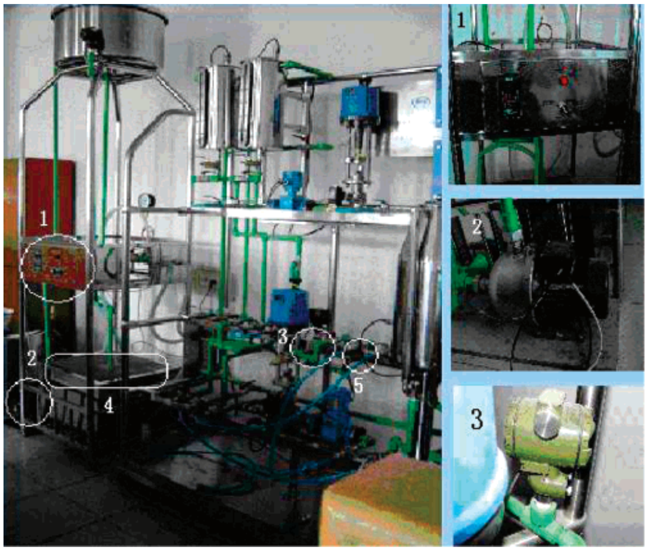

Figure 3. Constant pressure water supply system.

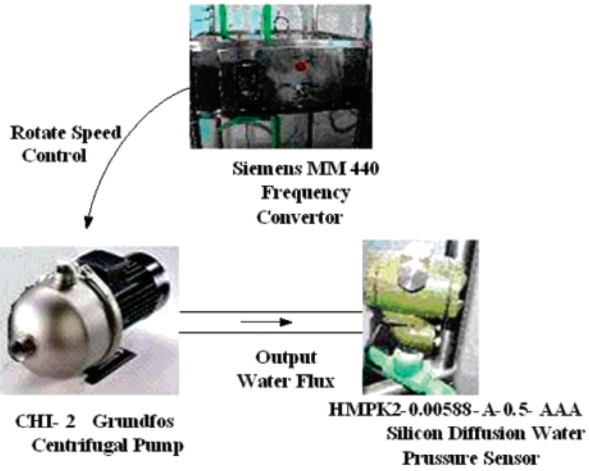
Rotate Speed Control
Siemens MM 440 Frequency Convertor
Output Water Flux
CHI- 2 Grundfos Centrifugal Pump
HMPK2-0.00588-A-0.5-AAA Silicon Diffusion Water Prussure Sensor

Figure 4. Signal and water flow chart.

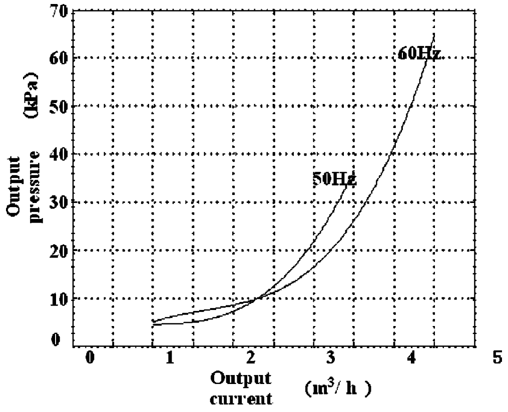

Figure 5. Characteristics curves of Grundfos CH_2-typed centrifugal pump.

(1) CNMPC (conventional NMPC),[6] which is based on the NAARX model (which is a special case of the NARMAX models) defined as

$$y_m(t) = \sum_{i=1}^{l} f_i(y(t-i)) + \sum_{i=1}^{n} g_i(u(t-i)) + \omega(t) \quad (30)$$

The functions $\{f(\cdot)\}$ and $\{g(\cdot)\}$ are scalar nonlinearities; l and n are input and output memories, respectively, and $\omega(t)$ is the

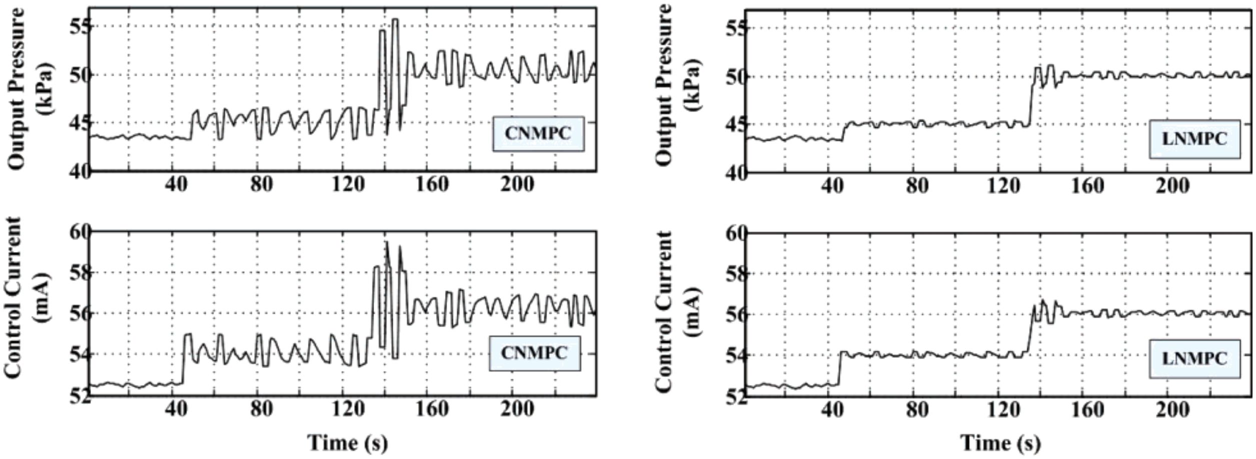

Figure 6. Control performances of tracking 35.4−43.4 KPa.

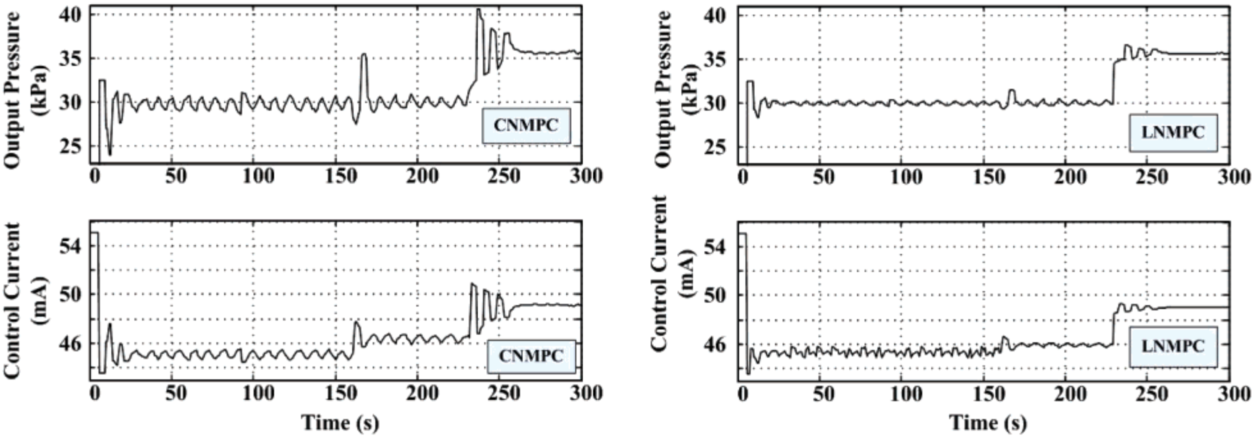

Figure 7. Control performances of tracking 30.0−35.6 KPa.

external white noise. This approach uses nonlinear programming to handle input magnitude constraints.

(2) LNMPC, which is the multistep Volterra−Laguerre model-based NMPC (see eq 20).

In this section, statistical experiments will be performed on a practical industrial system to further validate the feasibility and superiority of LNMPC.

The constant-pressure water supply system is shown in Figure 3. The executer is the Siemens MM440-type frequency converter, and the controlled plant is the Grundfos CHI-2 type centrifugal pump. The signal and water flowchart are shown in Figure 4. The pressure sensor transforms the output water pressure to a standard voltage signal (1−5 V). Through A/D transmission, the signal is sent to the serial port of a personal computer (PC). The controller in the PC calculates the control signal and sends it in the standard current signal (4−20 mA), through D/A transmission, to the Siemens frequency converter to control the rotation frequency of the Grundfos centrifugal pump. As shown in Figure 5, there is a strong nonlinearity between the output pressure and water current. Moreover, the nonlinear dynamics change along with the variations of the rotation frequency of the pump. There is a time delay in the water supply system; this delay is influenced by the variations of the flux velocity. Fortunately, experiments verify that this plant is a FMNS, according to the definition in Boyd and Chua;[4] thus, the current problem becomes the task of controlling a high-order FMNS plant with uncertainty and variational time delay, as shown in Figure 3. According to sections 1 and 2, this problem can be expected to be solved by our proposed algorithm.

The control performances of LNMPC are compared with CNMPC in Figures 6 (tracking the double-step signal 35.4−43.4 KPa), 7 (tracking the double-step signal 30.0−35.6 KPa), and 8 (tracking the double-step signal 45.0−50.0 KPa). Robustness to the external disturbance is also examined at the 160th second, as shown in Figure 7, where the opening of the hand valve 5 in Figure 3 is increased by 20%. The parameters in Figures 6 and 7 are set as follows:

For CNMPC: $P = 7$, $M = 1$, forgetting factor $\lambda = 0.7$, control signal weighting $r = 0.9$, the softening parameter $\alpha = 0.2$, $f_i(\cdot)$, $g_i(\cdot)$ in eq 30 are set to be polynomials with $l = 3$, $n = 5$.

For LNMPC: $P = 7$, $M = 3$, initial forgetting factor $\lambda_0 = 0.6$, $\alpha = 0.2$, $p = 0.74$, $N = 7$, $r = 0.9$.

Note that, in LNMPC, p is optimized according to Campello and co-workers;[12,13] N and r then are selected to guarantee the stability based on Theorem 1. The other parameters of CNMPC and LNMPC are almost the same, except for the difference values of the control horizon M, which is intended to show the merits of multistep control. To further examine the capability of the proposed algorithm, statistical experiments are conducted to track the double-step 45.0−50.0 KPa.

(1) For CNMPC, we set $P = 8$ and $M = 1$, and we select the input and output memories from the sets {4, 5, ..., 9} and {3, 4, 5, 6}, respectively. We also conduct 24 experiments.

(2) For LNMPC, we set $\lambda_0 = 0.98$ and select N and M from the sets {5, 6, ..., 10} and {2, 3, 4}, respectively. The parameters p and r are adjusted according to Theorem 1, with 18 experiments being conducted.

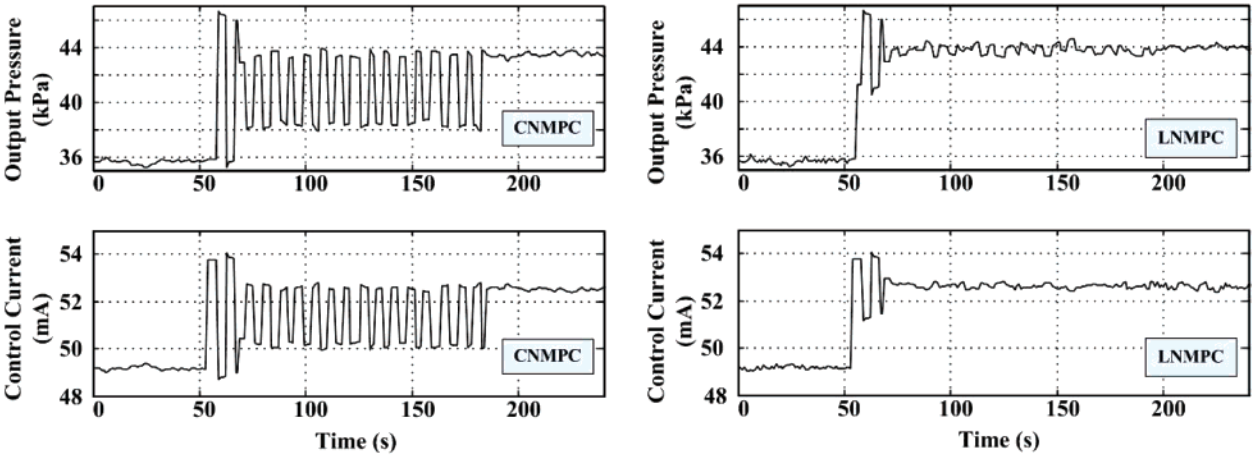

Figure 8. Control performances of tracking 45.0−50.0 KPa.

Table 1. Statistical Comparison of CNMPC and LNMPC Tracking 45.0−50.0 KPa

	settling time (s)	overshoot (KPa)	steady-state error (KPa)
CNMPC Control Index (24 Experiments)			
optimal input memory, output memory	6, 4		
expected value	165	±8.8	±2.7
optimal value	136	±6.3	±2.3
LNMPC Control Index (18 Experiments)			
optimal *N*, *M*, *p*	7, 3, 0.68		
expected value	23	±1.7	±0.8
optimal value	18	±1.5	±0.5

The statistical results, such as the expected and optimal settling time, overshoot, and steady-state errors are shown in Table 1, with optimal parameters set as $N = 7$, $M = 3$, $p = 0.68$, and their corresponding performances in Figure 6.

Experimental results in Figures 6−8 show the advantages of LNMPC over CNMPC, such as better transient performance, smaller steady-state errors, and robustness to the external disturbance. The underlying reasons are given as follows:

(1) With a multistep control horizon, the inner model of the LNMPC can predict the dynamics of the controlled plant more accurately, and it can enlarge the closed-loop region according to Theorem 1.

(2) The plant is a FMNS with variational time delay and uncertainty, which is very suitable to be approached via the Volterra−Laguerre model;

(3) The CNMPC uses nonlinear programming to handle input constraints, which cannot guarantee convergence to the objective function global optimum for non-convex problems, whereas the current analytic solution controller ensures convergence to the global minimum objective function value.

In regard to the computational complexity, although its Buchberger Algorithm, which calculates the Groebner bases, will require some time, it is not necessary for LNMPC to use nonlinear programming, which is an even more time-consuming method, to determine the global optimum. In addition, the present model can decrease both the number of model coefficients and the noise-induced parameter variation. Therefore, LNMPC is more efficient than CNMPC when applied to FMNSs with uncertainty and variational time delay (such as the example considered in this section).

6. Summary and Conclusions

The extension of the control horizon can help nonlinear model predictive control (NMPC) algorithms address complex nonlinearities. However, this extension is not easy to realize, because the online minimization of the objective function always leads to a set of complex polynomial equations, including the input of coupling items. The receding horizon optimization also causes extra difficulties for the closed-loop stability analysis. With these two problems in mind, we present a general method to extend the control horizon with the help of the Groebner basis technique. Sufficient conditions for its closed-loop stability are also provided, in the sense that the input and output series are both mean-square-bounded. The control horizon extension developed is applicable to a wide range of NMPCs based on open-loop stable discrete empirical models (NARMAX, NARX models, etc.). Finally, the proposed multistep control algorithm is demonstrated on an industrial water supply system with significant improvements in the transient performance and robustness to the system uncertainties.

Appendix A. Calculation of the Groebner Basis Using the Buchberger Algorithm

First, because its definition is not straightforward, we introduce some preliminary conceptions of the Groebner basis.

Definition 2 (Polynomial Ring).[28] The set of all polynomials $f = \sum_{\alpha} a_{\alpha} x^{\alpha}$ with coefficients in a field $\mathbf{K}$ is denoted as $\mathbf{K}[x_1, ..., x_n]$, which is a polynomial ring. In addition, the n-dimensional affine space over $\mathbf{K}$ is defined to be the set $\mathbf{K}^n = \{(a_1, ..., a_n): a_1, ..., a_n \in \mathbf{K}\}$.

■

Definition 3 (Ideal).[28] A subset $\mathbf{I} \subset \mathbf{K}[x_1,\cdots,x_n]$ is an ideal if it satisfies the following criteria:

(a) $0 \in \mathbf{I}$,

(b) If f, $g \in \mathbf{I}$, then $f + g \in \mathbf{I}$, and

(c) If $f \in \mathbf{I}$ and $h \in \mathbf{K}[x_1, ..., x_n]$, then $h \cdot \in \mathbf{I}$.

■

If $f_i \in \mathbf{K}[x_1, ..., x_n] (i = 1, ..., s)$, we denote that

$$\langle f_1,...,f_s \rangle \triangleq \{\textstyle\sum_{i=1}^{s} h_i f_i : h_1,...,h_s \in \mathbf{K}[x_1, ..., x_n]\}$$

then a crucial fact is that $\langle f_1, ..., f_s \rangle$ is an ideal.

Definition 4 (Leading Term). Given a nonzero polynomial $f \subset \mathbf{K}[x]$, let $f = a_0 x^m + a_1 x^{m-1} + \cdots + a_m$, with $a_0 \neq 0$. We then say that $a_0 x^m$ is the leading term of f and is written as $LT(f) = a_0 x^m$.

■

Lexicographic Order. The order of relative importance that should be used to illustrated Definition 4, e.g. if x_1 "is more important than" x_2, then $LT(x_1^2 x_2^3 + x_2^6) = x_1^2 x_2^3$.

Definition 5 (Groebner Basis).[28] Fix a monomial order. A finite subset $\{g_1, ..., g_s\}$ of an ideal $\mathbf{I}$ is said to be a Groebner basis if $\langle LT(g_1), ..., LT(g_s)\rangle = \langle LT(\mathbf{I})\rangle$.

■

The Groebner basis technique can be applied to determine all common solutions in $\mathbf{K}^n$ of a system of polynomial equations $f_1(x_1, ..., x_n) = \cdots = f_s(x_1, ..., x_n) = 0$. Let $x_i = \Delta u(t + i - 1|t)$ and $s = M$; the set of equations in eq 18 then can be solved by this technique. Buchberger's algorithm,[27] which is described in detail below, is used to determine the Groebner basis from a set of polynomial equations.

Definition 6. (S-polynomial).[28] Let $f, g \in \mathbf{K}[x_1, ..., x_n]$; the S-polynomial of f and g then is given as the combination

$$S(f, g) = \frac{\mathrm{LCM}(LT(f),LT(g))}{LT(f)} \cdot f - \frac{LCM(LT(f),LT(g))}{LT(g)} g \quad (31)$$

where LCM denotes the least common multiple.

■

Definition 7.[28] The term $\overline{f}^F$ is the remainder, upon division of f by the ordered s-tuple $F = (f_1, ..., f_s)$.

■

Bearing in mind Definitions 1−7, it is natural to introduce Burchberger's algorithm, as described below.

Buchberger's Algorithm.[27] Let $\mathbf{I} = \langle f_1,\cdots,f_s\rangle \neq \{0\}$ be a polynomial ideal. A Groebner basis for $\mathbf{I}$ then can be constructed in a finite number of steps using the following algorithm:

```
Input: $F = (f_1, ..., f_s)$
Output: a Groebner basis $G = (g_1, ..., g_t)$ for $\mathbf{I}$, with $F \subset G$
  REPEAT
      $G' := G$
    FOR each pair $\{p,q\}$, $p \neq q$ in $G'$ DO
      $S := \overline{S(p,q)}^{G'}$
    IF $S \neq 0$, THEN $G := G \cup \{S\}$
  UNTIL $G = G'$
```

■

With the lexicographic order x_1 "is more important than" x_2 "is more important than" ⋯ "is more important than" x_n, the resulting Groebner basis polynomials have a equation that contains only x_n; thus, the equation set can be easily solved. (For more details, please refer to the function *Groebner* (⋯) of MATHEMATICA). Note that the Groebner basis generated by Buchberger's algorithm is neither a minimal nor unique Groebner basis; some additional operations should be implemented to obtain the reduced Groebner basis that is minimal and unique.

For example, the equations $x_1^2 + x_2^2 + x_3^2 = 1$, $x_1^2 + x_3^2 = x_2$, $x_1 = x_3$ with lexicographic order x_1 "is more important than" x_2 "is more important than" x_3 can be computed by Buchberger's algorithm, combined with additional operations to get the minimal and unique Groebner basis $\{g_1, g_2, g_3\}$, which can be easily solved as described below:

$$g_1 = x_1 - x_3 = 0$$

$$g_2 = -x_2 + 2x_3^{\ 2} = 0$$

$$g_3 = x_3^{\ 4} + 0.5x_3^{\ 2} - 0.25 = 0$$

Appendix B. Derivation of the Key Polynomial (eq 27) for Stability Analysis

According to Lemma 1, the transformation of the closed-loop system to the polynomial form of eq 22 is fairly important for stability analysis; thus, we will show the deviation of eq 27 as follows.

Note that each eigenvalue of A is $p \in (0, 1)$; thus, $\lim_{i\to\infty} A^i = \vartheta$, where $\vartheta_{N\times N}$ is a zero matrix, and it is reasonable to assume that

$$(I - \mathbf{A}z^{-1})^{-1} = \sum_{j-0}^{N_z} A^j z^{-j} \quad (32)$$

where N_ϵ is a positive integer that is sufficiently large.

Substituting eq 32 into eq 8 yields

$$L(t|t) = z^{-1}(I - \mathbf{A}z^{-1})^{-1}\, \mathbf{B}u(t|t) = \sum_{j=0}^{N_\epsilon} A^{\mathrm{j}}\mathrm{z}^{-}(j{+}1)\mathbf{B}u(t|t) \quad (33)$$

Define

$$\Gamma(z^{-1},i)_{N\times 1} \triangleq [\bar{\mathbf{A}}_{\mathbf{i}} + \sum_{j=0}^{N_\epsilon} A^{i+j}z^{-(j+1)}]\mathbf{B} \quad (34)$$

then

$$L(t + i|t) = \Gamma(z^{-1},i)u(t|t),(i = 1),...,P \quad (35)$$

and

$$y_{\mathrm{m}}(t + i|t) = c_0 + \mathbf{C}^{\mathrm{T}}\Gamma(z^{-1},i)u(t|t) + \Gamma^{\mathrm{T}}(z^{-1},i)\mathbf{D}\Gamma(z^{-1},i)u^2(t|t) \quad (36)$$

Define

$$\Psi_0 \triangleq [c_0, ..., c_0]^{\mathrm{T}}_{P\times 1} \quad (37)$$

Thus,

$$\Psi_1(z^{-1},P) \triangleq [\mathbf{C}^{\mathrm{T}}\Gamma(z^{-1},1),...,\mathbf{C}^{\mathrm{T}}\Gamma(z^{-1},P)]^{\mathrm{T}}_{P\times 1} \quad (38)$$

$$\Psi_2(z^{-1},P) \triangleq [\Gamma^{\mathrm{T}}(z^{-1},1)\mathbf{D}\Gamma(z^{-1},1),..., \Gamma^{\mathrm{T}}(z^{-1},P)\mathbf{D}\Gamma(z^{-1},P)]^{\mathrm{T}}_{P\times 1} \quad (39)$$

$$\mathbf{Y}_m(t + 1) = \Psi_0 + \Psi_1(z^{-1},P)u(t|t) + \Psi_2(z^{-1},P)u^2(t|t) \quad (40)$$

Assume that

$$y(t) - y_{\mathrm{m}}(t) = \omega(t) \quad (41)$$

Then, $\partial J/\partial\Delta U(t|t) = 0$ (see eq 17) yields

$$\begin{aligned} &2\Psi_2^{\mathrm{T}}(z^{-1},P)\Psi_2(z^{-1},P)\Delta u^3(t) + \\ &3\Psi_2^{\mathrm{T}}(z^{-1},P)\Psi_1(z^{-1},P)\Delta u^2(t) + \\ &2[-2\Psi_2^{\mathrm{T}}(z^{-1},P)Y_{\mathrm{r}}(t + 1) + 2\Psi_2^{\mathrm{T}}(z^{-1},P)\Psi_0 + \\ &\Psi_1^{\mathrm{T}}(z^{-1},P)\Psi_1(z^{-1},P) + r]\Delta u(t) + \\ &[(1 - z^{-1})\Psi_1^{\mathrm{T}}(z^{-1},P)]\Omega\omega(t) - \\ &\Psi_1^{\mathrm{T}}(z^{-1},P)(1 - z^{-1})([z^{-(P-1)},\cdots \\ &z,1]^{\mathrm{T}}y_{\mathrm{r}}(t + P) - \Psi_0) = 0 \quad (42) \end{aligned}$$

Substituting our proposed multistep NMPC control law (eq 20) into eq 42 yields eq 27 with

$$T(z^{-1}) = -\frac{\alpha_\eta}{\alpha_3} \cdot \Psi_2^{\mathrm{T}}(z^{-1},P)\Psi_2(z^{-1},P) \qquad (43a)$$

$$H_0(z^{-1}) = -\Psi_1^{\mathrm{T}}(z^{-1},P)(1 - z^{-1})[z^{-(P-1)},...,z,1]^{\mathrm{T}} \qquad (43b)$$

$$D_0(z^{-1}) = [(1 - z^{-1})\Psi_1^{\mathrm{T}}(z^{-1},P)]\mathbf{\Omega} \qquad (43c)$$

The counterparts of the traditional single-step control algorithm (eq 19) are

$$T(z^{-1}) = 3\Psi_2^{\mathrm{T}}(z^{-1},P)\Psi_1(z^{-1},P)s_3 - 2\Psi_2^{\mathrm{T}}(z^{-1},P)\Psi_2(z^{-1},P)s_2$$
$$H_0(z^{-1}) = -\Psi_1^{\mathrm{T}}(z^{-1},P)(1 - z^{-1})[z^{-(P-1)},...,z,1]^{\mathrm{T}}s_3$$
$$D_0(z^{-1}) = [(1 - z^{-1})\Psi_1^{\mathrm{T}}(z^{-1},P)]\mathbf{\Omega}s_3 \qquad (44)$$

Because of their complexities, the detailed expressions of $T_i(z^{-1})$ (for $i = 1, ..., \mu - 1$), $D_0(z^{-1})$, and $\mathrm{H}_0(z^{-1})$ are omitted.

Acknowledgment

H. T. Zhang acknowledges the support of National Natural Science Foundation of China (NNSFC) under Grant No. 60704041, and the Natural Scientific Founding Project of Huazhong (Central China) University of Science and Technology under Grant No. 2006Q041B. H. X. Li acknowledges the support of RGC of HongKong SAR (CityU: 116905).

Nomenclature

$\mathbf{L}_2(\mathbb{R}_+)$ = square integrable space in $[0, \infty)$
I = identity matrix
$u(t)$ = system input
$y(t)$ = system output
$\omega(t)$ = external noise
z^{-1} = one-step-backward shifting operator
P = prediction horizon
M = control horizon
$\Delta u(t + i|t)$ = predicted incremental control signal series
$y(t + i|t)$ = predicted output signal series
$e(t)$ = modeling error
α = softening parameter
N = Laguerre series truncation length
N_{v} = Volterra series truncation length
$L(t)$ = Laguerre functional bases vector
A,**B** = Laguerre system state matrices
p = Laguerre filter pole
$h_m(t_1,t_2,...,t_m)$ = mth-order Volterra kernel
C = first order coefficients matrix of Volterra−Laguerre Model
D = second order coefficients matrix of Volterra−Laguerre Model
Q,**R** = positive-definite symmetric weighting matrices

Greek Symbols

θ = extended coefficient vector of the Volterra−Laguerre Model
$\Phi(t)$ = extended state vector of the Volterra−Laguerre Model
$\lambda(t)$ = forgetting factor
$\mathbf{\Omega}$ = output rectifying vector
$\omega(t)$ = modeling error

Superscripts

T = transpose
as = asymptotically

Subscripts

r = reference
m = model
c = continuous time
v = Volterra series

Abbreviations

MPC = model predictive control
CNMPC = conventional nonlinear MPC
LNMPC = multistep Volterra−Laguerre model-based nonlinear MPC
FMNS = fading memory nonlinear system
NARMAX = nonlinear auto-regressive moving average with exogenous inputs
OFS = orthonormal functional series

Literature Cited

(1) Camacho, E. F.; Bordons, C. *Model Predictive Control*, 3rd Edition; Springer: New York, 2003.

(2) Allgower, F.; Zheng, A. *Nonlinear Model Predictive Control*; Birkhauser Verlag: Basel, Boston, Berlin, 2000.

(3) Henson, M. A.; Seborg, D. E. *Nonlinear Process Control*; Prentice Hall: Upper Saddle River, NJ, 1997.

(4) Boyd, S.; Chua, L. Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Trans. Circuits Syst.* **1985**, *32* (11), 1150−1161.

(5) Schetzen, M. *The Volterra and Wiener Theories of Nonlinear Systems*; Wiley: New York, 1980.

(6) Dumont, G. A.; Fu, Y. Non-linear adaptive control via Laguerre expansion of Volterra series. *Int. J. Adaptive Control Signal Process.* **1993**, *7* (5), 367−382.

(7) Parker, R. S. Nonlinear model predictive control of a continuous bioreactor using approximate data-driven models. *Proc. Am. Control Conf.* **2002**, 2885−2890.

(8) Sistu, P. B.; Bequette, B. W. Nonlinear model-predictive control: Closed-loop stability analysis. *AIChE J.* **1996**, *42* (12), 3388−3402.

(9) Datta, K. B.; Mohan, B. M. *Orthogonal Functions in System and Control*; World Scientific: Singapore, 1995.

(10) Wahlberg, B.; Mäkilä, P. M. On approximation of stable linear dynamical systems using Laguerre and Kautz Functions. *Automatica* **1996**, *32* (5), 693−708.

(11) Heuberger, P. S. C.; Van den Hof, P. M. J.; Bosgra, O. H. A generalized orthonormal basis for linear dynamical system. *IEEE Trans. Autom. Control* **1995**, *40* (3), 451−465.

(12) Campello, R. J. G. B.; Favier, G., do Amaral, W. C. Optimal expansions of discrete-time Volterra models using Laguerre functions. *Automatica* **2004**, *40* (5), 815−822.

(13) Campello, R. J. G. B.; do Amaral, W. C.; Favier, G. A note on the optimal expansion of Volterra models using Laguerre functions. *Automatica* **2006**, *42* (4), 689−693.

(14) Zervos, C. C.; Dumont, G. Y. Deterministic adaptive control based on Laguerre series representation. *Int. J. Control* **1988**, *48* (6), 2333−2359.

(15) Dumont, G. A.; Zervos, C. C.; Pageau, G. L. Laguerre-based adaptive control of pH in an industrial bleach plant extraction stage. *Automatica* **1990**, *26* (4), 781−787.

(16) Oliveira, G. H .C.; Amaral, W. C.; Favier, G.; Dumont, G. A. Constrained robust predictive controller for uncertain processes modeled by orthonormal series functions. *Automatica* **2000**, *36* (4), 563−571.

(17) Wang, L. P. Discrete model predictive controller design using Laguerre functions. *J. Process Control* **2004**, *14* (2), 131−142.

(18) Adel, M. E.; Makoudi, M.; Radouane, L. Decentralized adaptive control of linear interconnected systems based on Laguerre series representation. *Automatica* **1999**, *35* (11), 1873−1881.

(19) Zhang, H. T.; Chen, Z. H.; Wang, Y. J.; Qin, T.; Li, M. Adaptive predictive control algorithm based on Laguerre Functional Model. *Int. J. Adaptive Control Signal Process.* **2006**, *20* (2), 53−76.

(20) Mäkilä, P. M. Laguerre series approximation of infinite dimensional systems. *Automatica* **1990**, *16* (6), 985−995.

(21) Fu, Y.; Dumont, G. A. An optimum time-scaling for discrete Laguerre network. *IEEE Trans. Autom. Control* **1993**, *38* (6), 934−938.

(22) Wang, L.; Cluett, W. R. Optimal choice of time-scaling factor for the linear system approximation using Laguerre models. *IEEE Trans. Autom. Control* **1994**, *39* (7), 1463−1467.

(23) Doyle, F. J., III; Pearson, R. K.; Ogunnaike, B. A. *Identification and Control Using Volterra Models*; Springer: Berlin, 2001.

(24) Parker, R. S.; Doyle, F. J., III; Peppas, N. A. A model-based algorithm for blood glucose control in Type 1 diabetic patients. *IEEE Trans. Biomed. Eng.* **1999**, *46* (2), 148−157.

(25) Parker, R. S.; Doyle, F. J., III. Optimal control of a continuous bioreactor using an empirical nonlinear model. *Ind. Eng. Chem. Res.* **2001**, *40* (8), 1939−1951.

(26) Parker, R. S.; Heemstra, D.; Doyle, F. J., III; Pearson, R. K.; Ogunnaike, B. A. The identification of nonlinear models for process control using tailored ‘plant-friendly’ input sequences. *J. Process Control* **2001**, *11* (2), 237−250.

(27) Buchberger, B. Grobner bases: An Algorithmic Method in Polynominal Ideal Theory. In *Multidimensional Systems Theory*; Bose, N. K. Ed.; D. Reidel Publishing Company: Dordrecht, The Netherlands, 1985: pp 184−232.

(28) Cox, D.; Little, J.; Shea, D. Q. *Ideals, Varieties and Algorithms* [B], 2nd Edition; Springer−Verlag: New York, 1997.

(29) Wang, W. *Generalized Predictive Control and Applications* [B]; *Chinese Science Press*, 1998.

(30) Goodwin, G. C.; Sin, K. S. *Adaptive Filtering, Prediction and Control* [B]; Prentice Hall: Englewood Cliffs, NJ, 1984.

(31) Ljung, L. *System Identification: Theory for the User*, 2nd Edition; Prentice Hall: Englewood Cliffs, NJ, 1999.

Received for review March 15, 2007
Revised manuscript received August 30, 2007
Accepted August 31, 2007

IE0703944