Constrained Nonlinear MPC Using Hammerstein and Wiener Models: PLS Framework

Rohit S. Patwardhan, S. Lakshminarayanan, and Sirish L. Shah
Dept. of Chemical and Materials Engineering, University of Alberta, Edmonton, T6G 2G6, Canada

Methods of implementing an input-constrained, nonlinear, model-predictive controller in latent spaces using partial-least-squares (PLS)-based Hammerstein and Wiener models are discussed. For multiple-input, multiple-output (MIMO) systems, the PLS framework presents a viable alternative for identification and controller synthesis using Hammerstein and Wiener structures. The constraint mappings, which have to be taken into account during controller design in the PLS framework, are highlighted. PLS-based Wiener models are well suited for constrained control of nonlinear systems. The use of PLS-based Hammerstein models for control involves solution of a nonlinear program as a result of the constraint mapping. The proposed approach is demonstrated on a simulated pH-level control of an acid–base neutralization process.

Introduction

The underlying philosophy of model predictive control (MPC) consists of minimization of a performance objective function with respect to future input moves, over a finite time horizon. Simplicity of design combined with its ability to tackle realities such as constraints and interactions has helped MPC achieve its current popularity with the process industry. MPC usually relies on time-domain models such as step-response or impulse-response models.

Most chemical processes are inherently nonlinear. Researchers have proposed several ways to equip MPC with the capability of dealing with nonlinear processes. The approaches to nonlinear MPC (NLMPC) can be categorized into two groups: (1) those based on first-principles models of the process, and (2) those based on black-box models identified from input/output data.

If the MPC objective function is to be minimized using the first-principles, nonlinear model, a dynamic optimization problem has to be solved at every sampling instant. Biegler (1984) proposed approximation of the differential equations using orthogonal collocation in order to reduce the optimal control problem to a nonlinear program that could be solved using sequential quadratic programming (SQP). Patwardhan et al. (1990) extended Biegler's approach to the constrained nonlinear MPC problem. This technique requires selection of appropriate orders for the polynomial approximations and increases the size of the optimization problem manifold. Economou and Morari (1986) adopted the nonlinear internal model control (IMC) framework wherein the nonlinear model inverse was obtained by a Newton-type method. Li and Biegler (1988, 1989) improved the convergence properties of this method by including a relaxation parameter. Constraints were incorporated through a quadratic programming (QP) formulation.

Brengel and Seider (1989) proposed an NLMPC scheme using successive linearization of the model equations at each sampling interval. This involves analytical/numerical evaluation of the Jacobians and discretization. The linearized model is used for predictions and only a quadratic program has to be solved at each iteration. Garcia (1984) used the full nonlinear model for finding the effect of the past inputs and estimated disturbances on the output and the linearized model for predicting the effect of the future inputs. Thus the objective function remains quadratic and the computational burden is minimal. Gattu and Zafirou (1992) extended this method by incorporating state estimation using a Kalman filter for better disturbance rejection. They successfully applied this scheme to a semibatch polymerization reactor and an unstable open-loop process.

Patwardhan and Madhavan (1993) approximated the model equations using a second-order Taylor series expansion around the steady state. For processes showing highly nonlinear behavior (e.g., input multiplicities) the second-order approximation proves superior in comparison to a linear model that includes only first-order terms. Mutha et al. (1997) pro-

Correspondence concerning this article should be addressed to S. L. Shah.

posed a nonlinear MPC scheme for control of nonaffine systems based on a reinterpretation of the prediction equation as a Taylor series expansion.

In a novel strategy based on the first-principles model, Henson and Kurtz (1994) and de Oliveira et al. (1995) proposed feedback linearization of the nonlinear process, followed by linear MPC of the feedback linearized loop. Though the model used in MPC design is linear, the nature of the mappings results in the constraints being nonlinear. De Oliveira et al. (1995) proposed a suboptimal strategy in order to preserve the linearity of the constraints. Henson and Kurtz (1994) formulated the constraint mapping as an optimization step, but sacrificed optimality to maintain linearity of the constraints.

Fundamental models of large chemical processes involve a significant effort and rely on complete knowledge of the process, plant operating conditions, and parameters. This kind of knowledge is often scarce, and a practical alternative is the use of nonlinear identification procedures to obtain the nonlinear mapping from input/output data.

Bhat and McAvoy (1990) developed artificial neural network (ANN) models of the nonlinear plant under consideration and proposed use of the ANN models for prediction in MPC. Though ANN models are known to approximate any type of nonlinear behavior, the modeling effort is large and a difficult nonlinear program has to be solved at every instant in order to calculate the control moves.

Hernandez and Arkun (1993) used polynomial ARMA models to capture the nonlinearities; the models consider cross terms between past inputs and outputs, and the estimation problem is posed as a linear least-squares problem. The MPC objective function is nonlinear, with the polynomial ARMA model as the predictor. Doyle et al. (1995) and Manner et al. (1996) use second-order Volterra models to describe nonlinear behavior. The Volterra model considers the crossterms between the past inputs in the manner of convolution models. A large number of coefficients are required for modeling purposes, and the optimization step is a nonlinear program. Carlemann-Rugh linearization of the mechanistic process model enables estimation of the model parameters without resorting to experimentation.

Hammerstein and Wiener models are block-oriented nonlinear models in which a linear dynamic element is cascaded in series to a static nonlinear element. For Hammerstein models, the nonlinear block precedes the linear dynamic element; for Wiener models the order of the linear and the nonlinear blocks is reversed. Zhu and Seborg (1994) use a Hammerstein model to do unconstrained MPC of a univariate pH control process. Norquay et al. (1996, 1997) demonstrate the use of a Wiener-model-based MPC on the same pH process. Norquay et al. (1997) propose an ad hoc scheme for handling constraints.

In this article a novel approach to input-constrained nonlinear MPC using partial-least-squares (PLS)-based Hammerstein and Wiener models is presented. It combines a powerful statistical tool, PLS, and two elegant representations of nonlinear systems, namely Hammerstein and Wiener models, with an advanced multivariate controller, MPC. Conventional MIMO Hammerstein and Wiener models present numerical difficulties for multivariable control. For control purposes, solution of a set of nonlinear algebraic equations is involved and this can lead to numerical instability. The use of a PLS framework decomposes the modeling problem into a series of univariate problems in the *latent subspace*. The controller synthesis is carried out in the latent subspace and facilitates the use of a series of SISO Hammerstein and Wiener models, for MIMO processes. Kaspar and Ray (1992, 1993) first proposed the use of PLS-based dynamic modeling and controller design for linear unconstrained systems.

The main contribution of the article is to show that the proposed approach is effective in handling multivariable processes and does not compromise the optimality of the constraints. It is shown that, for Wiener models, the input constraints preserve their linearity in the latent space, and only a quadratic program has to be solved to find the optimal input moves. For the Hammerstein model the input constraints get transformed into a nonlinear region in terms of the latent variables, and a nonlinear program results. Henson and Kurtz (1994) and de Oliveira et al. (1995) faced a similar situation with their feedback linearization + MPC scheme, but they opted for a suboptimal feasible approach to keep the constraint set linear. The proposed constrained multivariable NLMPC strategies are demonstrated on a simulation case study involving the pH-level control of a nonlinear acid–base neutralization process.

Nonlinear Modeling Using PLS-Based Hammerstein and Wiener Models

Partial least squares: overview

PLS has established itself as a robust alternative to the standard linear least-squares technique in the analysis of correlated data. First proposed by Wold (1966), this technique has found application in many disciplines such as social sciences, engineering, chemistry, and medicine. In the field of chemical engineering, PLS has found applications in the area of process monitoring, modeling, fault detection, and control.

In PLS the input (U) and output (Y) data are expressed as a sum of rank-one outer products

$$U = t_1 p_1^T + t_2 p_2^T + \cdots + t_n p_n^T + E_{n+1} = TP^T + E_{n+1}$$
$$Y = v_1 q_1^T + v_2 q_2^T + \cdots + v_n q_n^T + F_{n+1} = VQ^T + F_{n+1}, \quad (1)$$

where n is the number of PLS dimensions. In the preceding representation, T and V denote the scores, henceforth referred to as the latent variables (LVs); P and Q represent the loading matrices for the U and Y blocks, respectively; and the first set of loading vectors p_1 and q_1 is obtained by maximizing the covariance between U and Y. Projection of the U and Y data onto the loading vectors gives the first set of scores

$$t_1 = Uj_1$$
$$v_1 = Yq_1 \quad (2)$$

where

$$j_1 = \frac{U^T t_1}{t_1^T t_1}.$$

A linear regression between the scores results in the following *inner* relationship.

$$\hat{v}_1 = t_1 b_1. \quad (3)$$

In Eq. 3 $\hat{v}_1 q_1^T$ may be interpreted as the part of Y data that has been predicted by the first PLS dimension; in doing so, the $t_1 p_1^T$ portion of U has been utilized. The preceding procedure is repeated on appropriate residuals from the U and Y blocks to find the remaining scores/loading vectors until a major portion of the covariance between U and Y has been explained. This procedure is known as the PLS-NIPALS algorithm and was first developed by Wold (1966). The PLS technique has also been cast into the singular value decomposition (SVD) framework (Wise, 1991).

From a practical point of view, PLS can be considered to be a technique that decomposes a multivariate regression problem into a series of univariate regression problems. The PLS-based regression model can be written as

$$\hat{Y} = \hat{V}Q^T + F_{n+1} = TBQ^T + F_{n+1}, \quad (4)$$

where $B = \mathrm{diag}(b_1, b_2, \ldots, b_n)$.

PLS-based dynamic models

If the U block were to include the past values of the inputs/outputs, then analysis based on PLS would yield a dynamic model of the form

$$Y = UC_{\mathrm{dyn}} + e. \quad (5)$$

With this approach, C_{dyn} can be interpreted as a matrix whose elements are the finite impulse response coefficients (Ricker, 1988) or multivariable autoregressive moving-average (ARMA) model terms (Qin and McAvoy, 1992). Kaspar and Ray (1992, 1993) developed PLS-based dynamic models based on filtering of input data. The major dynamic component in the data was explained by filtering the data, and then the standard PLS procedure was used to model the filtered data.

Lakshminarayanan et al. (1997a) proposed a dynamic extension based on a modification of the PLS inner relation. Instead of relating the input and output scores (t_i and v_i) using a steady-state model, a dynamic component such as ARX is used. This gives the dynamic analog of Eq. 4 as

$$\hat{Y} = G_1(t_1)q_1^T + G_2(t_2)q_2^T + \cdots + G_n(t_n)q_n^T + F_{n+1}. \quad (6)$$

Here G_i denotes the linear dynamic models (e.g., ARX) identified at each stage. In terms of the latent variables we can write

$$V(z^{-1}) = \begin{bmatrix} G_1(z^{-1}) & 0 & \cdots & 0 \\ 0 & G_2(z^{-1}) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & G_n(z^{-1}) \end{bmatrix} T(z^{-1}) = G(z^{-1})T(z^{-1}). \quad (7)$$

As per Figure 1, the relationship between the original variables is given by

$$Y(z^{-1}) = \{QG(z^{-1})P^{\dagger}\}U(z^{-1}). \quad (8)$$

Note that G is a diagonal transfer-function matrix (inner relationship), but P and Q are full matrices (loadings). As a result the overall relationship between U and Y is a full transfer-function matrix given by

$$G_o(z^{-1}) = QG(z^{-1})P^{\dagger}. \quad (9)$$

Thus, despite the diagonal nature of the inner relationship, the overall model in terms of the original variables is multivariable and does capture the interactions among different inputs and outputs.

Remark 1. The dynamic PLS models are particularly well suited for controller design, since in terms of the latent variable framework we have several univariate models that can be used to design controllers based on SISO theory. This presents a strong case for the design of controllers in latent space. Suitable pre- and postcompensators have to be used to transform the variables from the latent to the original space and vice versa. For the unconstrained case the MIMO controller design problem is thus reduced to several SISO design problems.

Remark 2. For square as well as nonsquare systems, pairing in terms of the latent variables is automatic. For $n < \min(n_u, n_y)$ there is a dimensionality reduction of the original system. For such cases the mapping from the original variables to the latent variables is not exact, and a subsequent loss in information results.

Remark 3. The PLS estimation procedure is nonlinear by itself. Hence accuracy of parameter estimates, variance of estimates, and so forth, are not easy to quantify. The goodness of a PLS model often has to be determined through cross-validation and residual analysis alone.

Extensions to nonlinear systems

For nonlinear systems a natural extension of the dynamic PLS approach proposed by Lakshminarayanan et al. (1997a) leads to the use of nonlinear models for explaining the inner

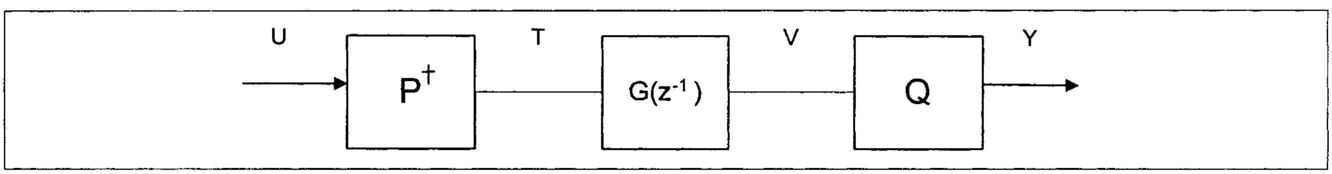

Figure 1. PLS-based dynamic models.

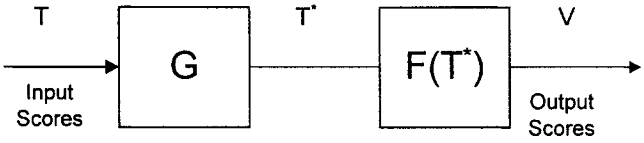

Figure 2. Wiener models in latent spaces.

relationship between the scores. In general, for any nonlinear modeling approach, like ANN, polynomial ARMA can be used. In this section we discuss Hammerstein and Wiener structures for modeling the inner relationship. The two approaches are best illustrated by Figures 2 and 3.

The Hammerstein model structure consists of a nonlinear algebraic element followed by a linear dynamic element. In contrast, the Wiener model configures the nonlinear static element following the linear dynamic element. See Haber and Unbehauen (1990) for a comparison of Hammerstein and Wiener structures with other input/output approaches to nonlinear identification. Eskinat et al. (1991) used the Hammerstein structure for modeling a simulated distillation column and an experimental heat exchanger. Lakshminarayanan et al. (1995) extended the canonical variate analysis technique, a subspace identification method, to the estimation of MIMO Hammerstein models. Verhagen and Westwick (1996) modified the MOESP family of subspace methods for identification of Hammerstein-type systems.

PLS-based Hammerstein models are obtained by relating the scores at each dimension using an SISO Hammerstein model. A polynomial function is used to capture the nonlinearity of the process. The order of the chosen polynomial is an indication of the degree of nonlinearity of the process. For an elaborate discussion of the PLS-based Hammerstein modeling approach the reader is referred to Lakshminarayanan et al. (1997a).

The inner relation between the input and output scores is given by

Hammerstein model:

$$t_i^* = f_i(t_i);\, v_i(z^{-1}) = G_i(z^{-1})t_i^*(z^{-1});\, i = 1, \ldots, n$$

Wiener model:

$$t_i^*(z^{-1}) = G_i(z^{-1})t_i(z^{-1});\, v_i = f_i(t_i^*);\, i = 1, \ldots, n. \quad (10)$$

Constrained NLMPC via PLS-Based Hammerstein and Wiener Models

In this section we discuss nonlinear model-predictive control using PLS-based Hammerstein and Wiener models. The standard MPC formulation is reviewed. Linear MPC based on dynamic PLS models is introduced with emphasis on the constraint mapping.

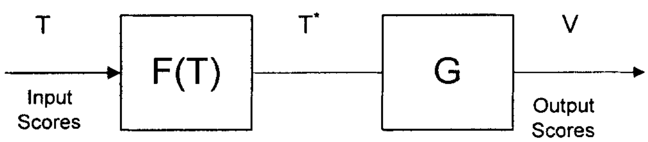

Figure 3. Hammerstein models in latent spaces.

MPC: unconstrained control law

The formulation of the multivariable MPC is discussed here. Consider a process with r outputs and m inputs. The model in step-response form is given by

$$\hat{y} = S\Delta u + f + d, \quad (11)$$

where

$$S = \begin{bmatrix} s_1 & 0 & & \cdots & 0 \\ s_2 & s_1 & 0 & \cdots & 0 \\ s_3 & s_2 & s_1 & 0 & 0 \\ \vdots & \vdots & & \ddots & \\ s_P & s_{P-1} & & \cdots & s_{P-M+1} \end{bmatrix}_{rP \times mM}$$

$$\Delta u = [\Delta u_1(k), \ldots, \Delta u_m(k), \ldots, \Delta u_1(k+M-1), \ldots, \Delta u_m(k+M-1)]^T$$

$$\hat{y} = [\hat{y}_1(k+1), \ldots, \hat{y}_r(k+1), \ldots, \hat{y}_1(k+P), \ldots, \hat{y}_r(k+P)]^T$$

$$f = [f_1(k+1), \ldots, f_r(k+1), \ldots, f_1(k+P), \ldots, f_r(k+P)]^T$$

$$d = [d(k+1), d(k+2), \ldots, d(k+P)]^T.$$

Each s_i is a matrix of dimension $r \times m$ consisting of the step-response coefficients of the process model. The second term on the righthand side denotes the free response of the system, that is, the response if there were no future inputs. This free response is expressed in terms of the past inputs and can be written as

$$f(k+i) = \sum_{j=i+1}^{N} s_j \Delta u(k-j+i) + s_{ss} u(k+i-N-1). \quad (12)$$

The disturbance vector d is estimated by subtracting the model output from the measured output as

$$d(k) = y(k) - \hat{y}(k) \quad (13)$$

$$d(k+i) = d(k); \quad i = 1, \ldots, P_2. \quad (14)$$

The objective function for MPC-type controllers considers the future deviations from the set point P moves into the future, and the differenced input M moves into the future, both squared and weighted. P is called the prediction horizon and M is known as the control horizon. Objective function:

$$J = (r - \hat{y})^T \Gamma (r - \hat{y}) + \Delta u^T \Lambda \Delta u \quad (15)$$

where $r = [r(k+1),\ r(k+2),\ \ldots,\ r(k+P)]^T$ the set point vector. The weighting matrices Γ and Λ are positive semidefinite weighting matrices usually diagonal in nature, with the individual weights indicating the relative significance of the different inputs/outputs. The sequence of steps leading to the minimization of the objective function with respect to the future input moves Δu is outlined below:

$$J = (r - S\Delta u - f - d)^T \Gamma (r - S\Delta u - f - d) + \Delta u^T \Lambda \Delta u.$$

The optimal control moves, obtained by minimizing this objective, are given by

$$\Delta u = (S^T \Gamma S + \Lambda)^{-1} S^T \Gamma (r - f - d). \quad (16)$$

In receding-horizon fashion only the first of the M moves is implemented:

$$\Delta u(k) = [I \; 0 \; \cdots \; 0](S^T \Gamma S + \Lambda)^{-1} S^T \Gamma (r - f - d). \quad (17)$$

Constrained Case. A real plant has to work within certain physical limitations, for example, a valve can handle only a particular range of flow rates, and market forces that result in rigid quality requirements on the process outputs. A model-predictive controller is well suited to handle the constraints in an elegant manner. The constraints on the inputs, slew rates, and outputs can be grouped together into a single *linear matrix inequality*. The objective function is then minimized subject to linear constraints on the future inputs. A quadratic program is solved at every sampling instant to arrive at the input values to be implemented on the plant. One cannot say *a priori* which set of constraints may be active at a sampling instant. In some cases only the slew-rate constraints are active; in other cases, both slew-rate and input constraints are active. Although the constrained solution is obtained by solving a set of linear equations, the changing nature of the active constraint set renders the control law *nonlinear*.

The quadratic program solved at every instant is

$$\underset{\Delta u}{\text{Min}}\, J = \underbrace{\Delta u^T (S^T \Gamma S + \Lambda)}_{\triangleq He/2} \Delta u - 2 \underbrace{(r - f - d)^T \Gamma S}_{\triangleq -c^T} \Delta u, \quad (18)$$

(18)

subject to $A\Delta u + B \geq 0$. Let (A^*, B^*) denote the active constraint set and λ be the corresponding Lagrangian multipliers. Therefore the constrained solution is obtained by solving the following set of linear equations

$$\begin{bmatrix} He & A^{*T} \\ A^* & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \lambda \end{bmatrix} = \begin{bmatrix} -c \\ -B^* \end{bmatrix}. \quad (19)$$

The overall control law can be expressed as

$$\Delta u = F[u(k-1), \ldots, u(k-N), y(k), r(k)]. \quad (20)$$

PLS-based constrained MPC of linear systems

Next we concern ourselves with the design of a constrained model-predictive controller using PLS-based dynamic models for linear systems. In this case, the MPC objective function is based on the linear dynamic model between t (inputs) and v (outputs):

$$\underset{\Delta t}{\text{Min}}\, J_L = (\Delta t)^T (S^T \Gamma S + \Lambda) \Delta t - 2(r - f - d)^T \Gamma S \Delta t, \quad (21)$$

subject to $A' \Delta t + B' \geq 0$.

In this section we will discuss the constraint mapping when we pose the problem in terms of the latent variables, that is, how to arrive at A', B'. The individual transformations relating the original to the latent variables are given by

$$t = P^{\dagger} u$$
$$y = Qv,$$

where $P^{\dagger}$ is the generalized inverse of P.

Consider a 2×2 system that is modeled using two PLS dimensions. The constraints in the original variables are decoupled if we restrict ourselves to input and slew-rate constraints, for example, constraints on u_1 and u_2 are independent of each other. For the sake of brevity only the two-variable case is illustrated here, but the concepts presented can be quite easily extended to the higher dimensional case. To illustrate our point geometrically, let us look at the region where u_1 and u_2 are required to remain.

This region can be represented by the inequality

$$u_{\min} \leq u \leq u_{\max} \Leftrightarrow$$
$$u_{\min} \leq Pt \leq u_{\max}.$$

Expanding we get

$$\begin{bmatrix} u_1^{\min} \\ u_2^{\min} \end{bmatrix} \leq \begin{bmatrix} p_{11} t_1 + p_{12} t_2 \\ p_{21} t_1 + p_{22} t_2 \end{bmatrix} \leq \begin{bmatrix} u_1^{\max} \\ u_2^{\max} \end{bmatrix}. \quad (22)$$

This region is shown in Figure 4. Thus the constraints get *coupled* in the latent space. The rate (Δu) constraints get affected similarly. The main outcome of posing the constraints in the latent space is that the problem gets coupled through the constraints in spite of the dynamics being decoupled in the latent space. Therefore a multivariate approach to controller synthesis is needed to handle the constraints. In this situation SISO theory cannot be used to design controllers in the latent space. If the output constraints are being used, then the constraints are coupled irrespective of the transformations. Often, the output constraints are excluded since they often lead to infeasibility problems in the optimization step. For a more detailed discussion of linear MPC design via PLS-based dynamic models the reader is referred to Lakshminarayanan et al. (1997b). Figure 5 illustrates the implementation MPC in a PLS framework.

Control strategy for Wiener- and Hammerstein-based MPC

The inherent structure present in Hammerstein and Wiener models leads to a very intuitive control strategy. In both cases, a linear MPC is designed, with the linear dynamic element

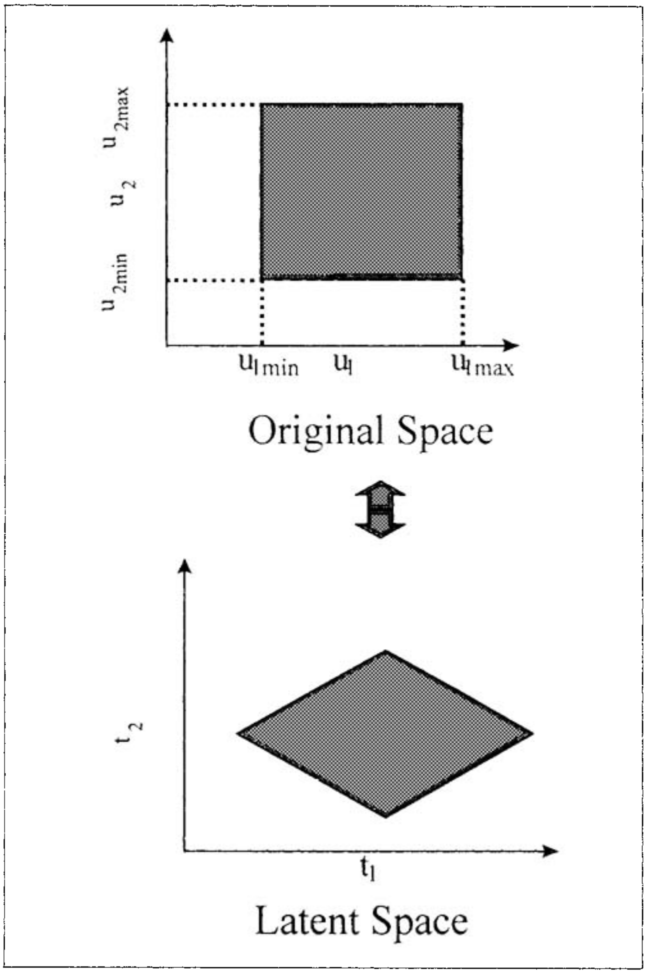

Figure 4. Constraint regions in the original and latent space.

serving as the model and the nonlinear operator is inverted. The order in which the linear MPC and the inversion operation is carried out depends on the model structure (see Figures 6 and 7). Assuming that the nonlinearity is well captured in the region of operation, this kind of control strategy results in an approximately *feedback linearized* system.

Remark 4. It should be noted here that the inversion of nonlinear operator F is fairly straightforward in the latent space since it is a decoupled system, that is, $F(T) = [f_1(t_1), f_2(t_2), \ldots, f_n(t_n)]$. Each f_i is a polynomial equation in a single variable t_i. Hence inversion is guaranteed, since robust numerical techniques are available for solving polynomial

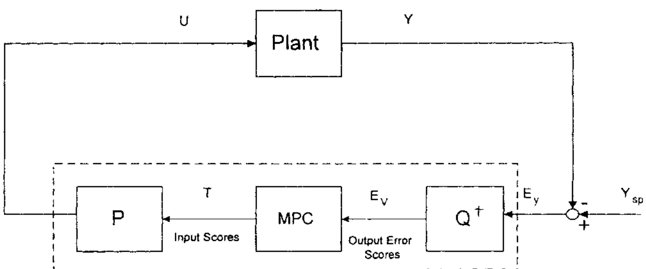

Figure 5. Linear MPC based on dynamic PLS models.

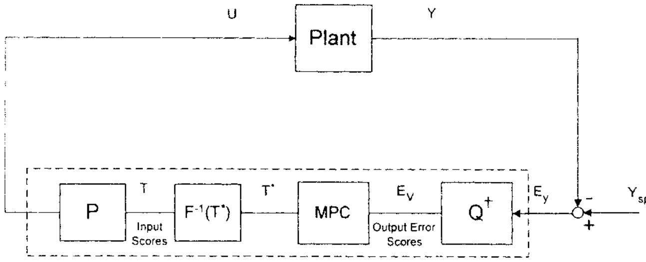

Figure 6. Control strategy for PLS-based Hammerstein models.

equations in one variable. In general the polynomial will have multiple roots; we choose the smallest real root. In this regard it is safe to choose a polynomial with an odd degree to ensure the presence of at least one real root.

Remark 5. In the ideal case of no model plant mismatch, the resulting feedback system is linear and linear stability analysis can be used for assessing the closed-loop stability of the nonlinear system. However, in practice, model plant mismatch is inevitable, and nonlinear stability analysis will have to be carried out for the general case.

In the remaining discussion we concern ourselves with the application of constrained MPC on nonlinear systems modeled using Hammerstein and Wiener models.

Constraint mapping

Hammerstein Models. The MPC objective function is based on the linear dynamic element between t^* (inputs) and v (outputs).

$$\underset{\Delta t^*}{\mathrm{Min}}\, J_H = (\Delta t^*)^T (S^T \Gamma S + \Lambda) \Delta t^* - 2(r - f - d)^T \Gamma S \Delta t^*, \tag{23}$$

subject to $g(t^*) \geq 0$. For the sake of brevity, we use t to denote the $n \times 1$-D latent space vector.

The nature of the transformations involved leads to the transition of the linear constraints in the original space to nonlinear constraints in the latent space. The following steps illustrate our point.

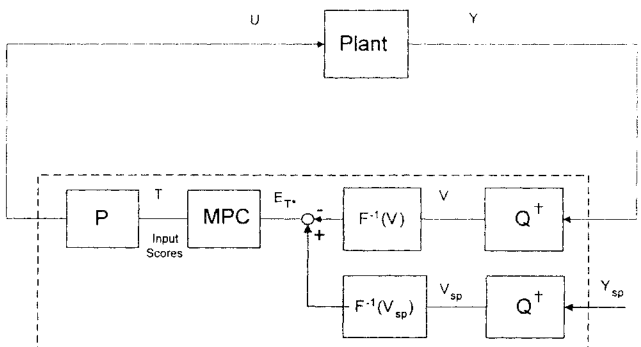

Figure 7. Control strategy for PLS-based Wiener models.

$$A\Delta u + B \geq 0 \Leftrightarrow$$
$$A'\Delta t + B \geq 0 \Rightarrow$$
$$\underbrace{A'\begin{bmatrix} F^{-1}(t_k^*) - F^{-1}(t_{k-1}^*) \\ F^{-1}(t_{k+1}^*) - F^{-1}(t_k^*) \\ \vdots \\ F^{-1}(t_{k+M-1}^*) - F^{-1}(t_{k+M-2}^*) \end{bmatrix} + B'}_{\triangleq g(t^*)} \geq 0. \quad (24)$$

The transformation from u to t is linear and is given by

$$u(k) = Pt(k)$$
$$A' = A[\text{diag}(P\ P\ \cdots\ P)]_{mM \times nM}$$

The transformation from t^* to t: $t_i^* = f_i(t_i)$, however, is nonlinear. The inverse operator is not one-to-one and hence the constraint mapping is nonunique.

From Eq. 24, it is clear that computation of the control moves involves solution of a nonlinear program at every instant. The nonlinear program comprises a quadratic objective function and a nonlinear constraint set. The nonlinearity of the constraints arises due to the particular structure of the Hammerstein model.

The steps involved in the implementation of the Hammerstein-based NLMPC scheme (see Figure 6) are enumerated below:

1. At sampling instant k measure the output $y_m(k)$ and the set point $r(k)$.
2. Transform the output and set point in terms of latent variables using the following relations: $\upsilon(k) = (S_y Q)^{-1} y(k)$; $\upsilon_{sp}(k) = (S_y Q)^{-1} r(k)$. Send the transformed variables to the PLS-based MPC.
3. The controller is based on the linear dynamic part and will calculate inputs in terms of $t^*(k)$. This is done by solving the nonlinear program using a suitable optimization tool. The initial guesses for the solver are provided by equating the initial values to the previous solution: $t_i^* = t^*(k-1)$. For $k = 1$, the initial guess is set to zero.
4. The inverse operator is used to map these values onto the corresponding $t(k)$: $t(k) = F^{-1}(t^*(k))$.
5. The inputs in terms of the latent variables are mapped onto the inputs in terms of original variables: $u(k) = S_u Pt(k)$.
6. Implement the calculated inputs, $u(k)$, on the plant.

Here S_u and S_y are the input and output scaling matrices, respectively.

Wiener Models. The nonlinearity of the constraints can be avoided if a Wiener-model structure is chosen. The Wiener-model structure succeeds in preserving the linearity of the constraints in spite of the nonlinear mappings involved. However, this is not true if output constraints are also involved. For the present purposes we restrict ourselves to rate and amplitude constraints only, since output constraints often result in infeasibility problems. In this case the MPC objective function is based on the linear dynamic model between t (inputs) and t^* (outputs):

$$\underset{\Delta t}{\text{Min}}\, J_W = (\Delta t)^T (S^T \Gamma S + \Lambda) \Delta t - 2(r - f - d)^T \Gamma S(\Delta t), \quad (25)$$

subject to $A'\Delta t + B \geq 0$.

The linearity of the constraints is preserved since the nonlinearity is now on the output instead of the input and the transformation from u to t is a linear one:

$$A\Delta u + B \geq 0 \Leftrightarrow$$
$$A'\Delta t + B \geq 0. \quad (26)$$

The relationship between A and A' remains the same as before. The geometry of this constraint set is the same as in Figure 4.

Thus a simple change in the model structure results in high gains in terms of the computational complexity of the optimization problem. Only a quadratic program has to be solved at every instant. In cases where the Wiener modeling approach is applicable, it offers distinct advantages for design of a constrained controller over other NLMPC methods based on black-box models.

Remark 6. In the latent space, Wiener models preserve the linearity of the input constraints. Hammerstein models result in the original linear constraints being mapped to a nonlinear constraint region.

The steps involved in the implementation of the Wiener-based NLMPC scheme (see Figure 7) are enumerated below:

1. At sampling instant k, measure the output $y(k)$ and the set point $r(k)$.
2. Transform the output and set point in terms of latent variables using the following relations: $\upsilon(k) = (S_y Q)^{-1} y(k)$; $\upsilon_{sp}(k) = (S_y Q)^{-1} r(k)$.
3. Invert the nonlinear operator to express these values in terms of t^*: $t^*(k) = F^{-1}[\upsilon(k)]$; $t_{sp}^*(k) = F^{-1}[\upsilon_{sp}(k)]$. Send the transformed variables to the PLS-based MPC.
4. The controller is based on the linear dynamic part and calculates the inputs in terms of $t(k)$. This is done by solving the quadratic program using any standard QP solver.
5. The inputs in terms of the latent variables are mapped onto the inputs in terms of original variables: $u(k) = S_u Pt(k)$.
6. Implement the calculated inputs, $u(k)$, on the plant.

Discussion

The ability to handle output constraints is one of the main advantages of constrained MPC. The output constraints can be posed in the following manner for Hammerstein and Wiener models, respectively.

Hammerstein Models.

Original variables: $y_{\min} \leq \hat{y} \leq y_{\max}$

Latent variables: $y_{\min} \leq Q^{\dagger} \upsilon \leq y_{\max}$

$$y_{\min} \leq Q^{\dagger}\{S\Delta t^* + f + d\} \leq y_{\max},$$

where S, f, and d are obtained from the linear dynamic element between υ and t^*. The output constraints are linear in terms of the latent variables for PLS-based Hammerstein models.

Wiener Models.

$$\text{Original variables: } y_{\min} \le \hat{y} \le y_{\max}$$

$$\text{Latent variables: } y_{\min} \le Q^{\dagger} v \le y_{\max}$$

$$y_{\min} \le Q^{\dagger}\left\{F^{-1}\begin{bmatrix} t_{k+1} \\ t_{k+2} \\ \vdots \\ t_{k+P} \end{bmatrix}\right\} \le y_{\max},$$

where F is the nonlinear mapping between v and t. Thus for the Wiener models the output constraints are nonlinear in terms of the latent variables. These output constraints can be included in the optimization step along with the input constraints.

PLS is generally associated with dimensionality reduction. The main purpose of this work is not dimensionality reduction but to exploit the structural advantages of PLS-based models to do constrained control of a class of nonlinear systems. PLS is mainly used here to facilitate transformation of variables to a new basis space where the relationship between the input and output scores is decoupled.

In cases where there is dimensionality reduction the PLS model captures a large portion of the covariance between the inputs and outputs. In such cases the lower dimensional PLS model will still capture the dominant dynamics of the process. The dimensionality reduction will, however, affect the constraint mappings. When there is a dimensionality reduction, the P and Q matrices will be, in general, nonsquare. This will lead to a loss in information when the constraints are mapped from the original space to the latent space. But this problem can be resolved by using the full P and Q matrices (without dimensionality reduction) for constraint mapping alone.

Case Study: pH-Level Control of an Acid–Base Neutralization Process

The proposed approach was evaluated on a MIMO, pH-level control process. This example is an acid–base (HCl–NaOH) neutralization process performed in a single tank. pH neutralization processes are challenging control problems due to the inherent nonlinearity in the titration curve. The system description, the nonlinear process model, and the operating conditions are described in Henson and Seborg (1994). The level and the pH are the controlled variables that are manipulated using the acid and the base flow rates. The buffer flow rate serves as a disturbance. The nominal operating conditions of the process are:

Outputs: pH = 7.075, level = 14 cm

Inputs: acid = 16.6 mL/s, base = 15.6 mL/s, buffer = 0.6 mL/s.

Control of this process was attempted using a linear PLS-based MPC and compared with the nonlinear MPC approach based on Wiener/Hammerstein models as proposed here. A simulation model of the process was built in the

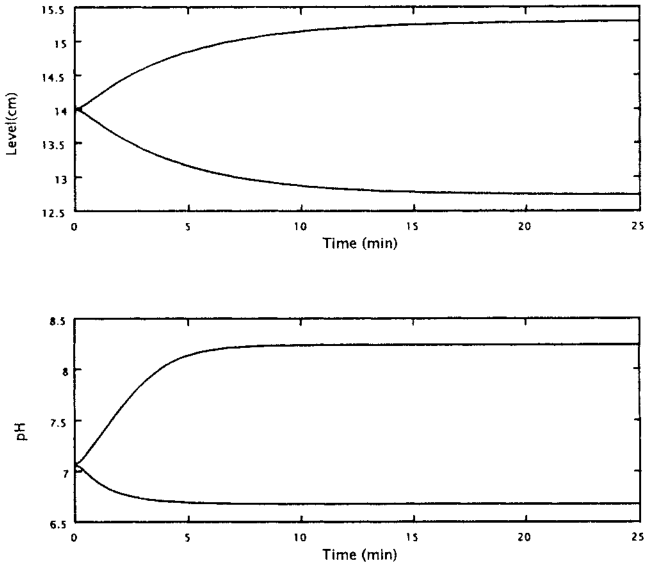

Figure 8. Open-loop simulation: level and pH responses to step changes in acid flow rate (±1 mL/s).

Simulink/Matlab environment. The open-loop simulations shown in Figure 8 indicate the nonlinear nature of the pH response. Although the pH response is fairly nonlinear in terms of the gains, it should be noted that the dynamics remain fairly unchanged. It is this feature that makes the pH system suitable for modeling using Hammerstein and Wiener models, which have a static nonlinear component. The identification experiments have to be designed to capture the nonlinearity of the system, that is, the magnitude of the input changes is of prime importance and will decide what kind of nonlinearities are excited. To capture the nonlinearity in the gain, the probing signals were designed to be low-frequency excitation spread over several levels of magnitude.

PLS-based, linear, and nonlinear models were estimated for the system using data collected from the identification runs. The model parameters are given below. Two PLS dimensions ($n = 2$) were used to explain the information in the data set. Thus there was no dimensionality reduction for this example. A fifth-order polynomial was required for the Hammerstein model, whereas a third-order polynomial proved sufficient for the Wiener model. The inner relationship for the second PLS dimension was linear for both methods.

Linear model

Scaling Matrices. The variables are normalized (unit variance, zero mean) by the scaling matrices:

$$S_u = \text{diag}(1.135, 1.0122), \qquad S_y = \text{diag}(1.2875, 1.3079).$$

The loading matrices are

$$P = \begin{bmatrix} 0.7220 & 0.6796 \\ -0.6919 & 0.7335 \end{bmatrix}, \qquad Q = \begin{bmatrix} 0.0347 & 1 \\ -0.9994 & 0.0081 \end{bmatrix}.$$

The linear dynamic parts were estimated as

$$G_1 = \frac{0.0267z^{-1} - 0.0094z^{-2}}{1 - 1.5885z^{-1} + 0.006z^{-2}},$$

$$G_2 = \frac{0.0435z^{-1} + 0.0539z^{-2}}{1 - 0.8794z^{-1} - 0.0556z^{-2}}.$$

Hammerstein model

Scaling Matrices. The variables are normalized (unit variance, zero mean) by the scaling matrices:

$$S_u = \text{diag}(1.135, 1.0122), \qquad S_y = \text{diag}(1.2875, 1.3079).$$

The loading matrices are

$$P = \begin{bmatrix} 0.7220 & 0.6796 \\ -0.6919 & 0.7335 \end{bmatrix}, \qquad Q = \begin{bmatrix} 0.0347 & 1 \\ -0.9994 & 0.0081 \end{bmatrix}.$$

The static nonlinear part is given by

$$t_1^* = 0.025t_1^5 + 0.1227t_1^4 - 0.0978t_1^3 - 0.5909t_1^2 + t_1$$

$$t_2^* = t_2.$$

The linear dynamic parts were estimated as

$$G_1 = \frac{0.161z^{-1} - 0.180z^{-2}}{1 - 0.8849z^{-1} + 0.0388z^{-2}},$$

$$G_2 = \frac{0.0458z^{-1} + 0.0522z^{-2}}{1 - 0.8744z^{-1} - 0.0601z^{-2}}.$$

The cross-validation of the Hammerstein model is shown in Figure 9. It is compared with a linear PLS-based model.

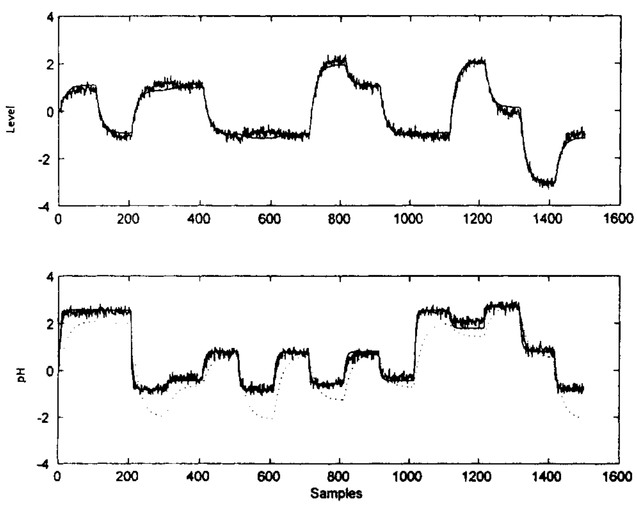

Figure 9. Cross-validation: Hammerstein model (---) vs. linear model (···) and actual data (——).

As can be seen from the figure, the Hammerstein model gives a good fit for the pH, whereas the linear model fails to capture the inherent nonlinearity for the pH, but does well with the level part.

Wiener model

The scalings chosen for the Wiener model were

$$S_u = \text{diag}(1.1989, 1.0202), \qquad S_y = \text{diag}(1.0664, 1.0252).$$

The loading matrices are given by

$$P = \begin{bmatrix} 0.6806 & 0.7103 \\ -0.7334 & 0.7039 \end{bmatrix}, \qquad Q = \begin{bmatrix} 0.0665 & 0.9981 \\ -0.9978 & -0.0611 \end{bmatrix}.$$

The linear dynamic parts were estimated as

$$G_1 = \frac{0.1740z^{-1}}{1 - 0.8754z^{-1}}, \qquad G_2 = \frac{0.1487z^{-1}}{1 - 0.432z^{-1} - 0.484z^{-2}}.$$

The static nonlinear parts were given by

$$v_1 = -0.1173t_1^{*3} - 0.3539t_1^{*2} + t_1^*$$

$$v_2 = t_2^*.$$

The cross-validation plots for the Wiener model are shown in Figure 10. The Wiener model gives a good fit for the pH part and appears to have captured the nonlinearity present in the system.

Three PLS-based controllers were evaluated: (1) Linear MPC; (2) Hammerstein-based NLMPC; and (3) Wiener-based NLMPC. For control schemes (1) and (3), the Matlab function *qp* was used to solve the quadratic program, whereas for (2) the nonlinear programming solver *constr*, available under the Optimization Toolbox in Matlab, was invoked. The sampling time was 15 s.

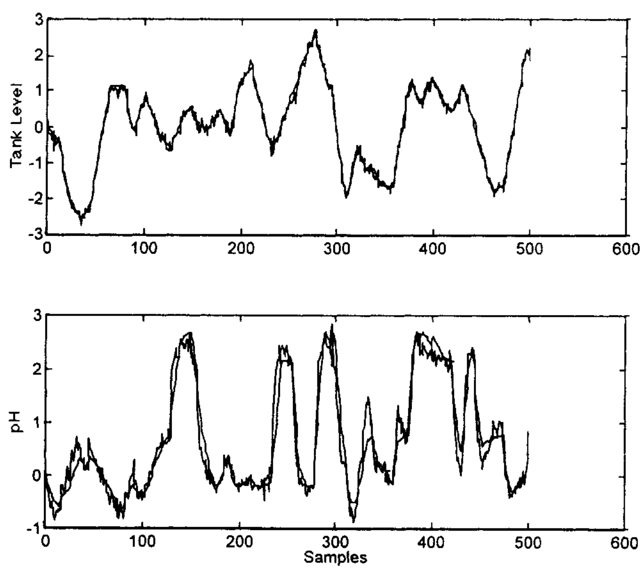

Figure 10. Cross-validation fit for the Wiener model.

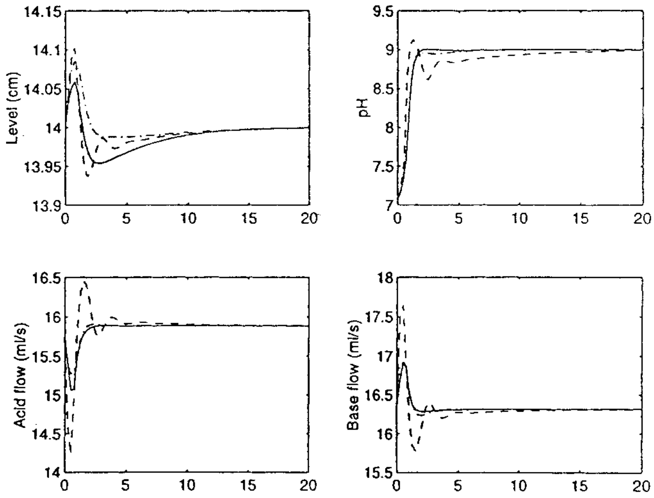

Figure 11. Comparison of (1) linear MPC (---), (2) NLMPC Hammerstein (——), and (3) NLMPC-Wiener (—·) for set point changes: pH-9, level = 14.

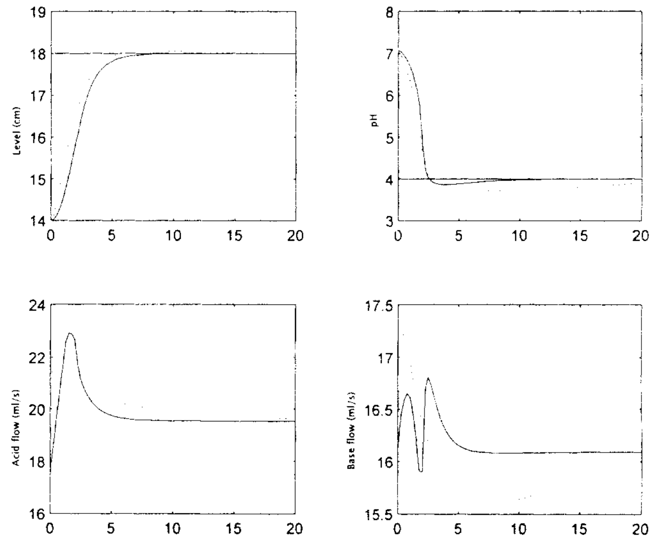

Figure 12. Tracking performance of NLMPC based on (1) Hammerstein models (···) and (2) Wiener models (——) for set point changes: pH = 4, level = 18.

Selection of Controller Tuning Parameters

$$P_2 = 15, \quad M_u = 2, \quad \Lambda = 0.5I, \quad \Gamma = I,$$

$$-1 \leq \Delta u_i(k) \leq 1, \quad -10 \leq u_i(k) \leq 10.$$

Figure 11 shows the performance of the three controllers in response to a set point change in pH (pH = 9). The Wiener-based NLMPC gives the best performance among the three controllers. The change in pH is achieved in a satisfactory manner with no overshoot in a settling time of 2 min. On the other hand, the linear MPC showed very poor performance in achieving the pH set point change. It appears to struggle with the nonlinearities in the system before reaching the set point in 15 min. The Hammerstein-based NLMPC showed good tracking properties, but its disturbance rejection for level was less than satisfactory. As opposed to the linear MPC, it gave smooth response in both the channels. Thus both the Hammerstein- and Wiener-based MPC schemes appeared to annul the nonlinearities in the system through feedback and gave responses similar to a linear system.

Figure 12 shows the tracking properties of control schemes (2) and (3) for set point changes in both level and pH. Clearly the Wiener-based NLMPC gives a superior performance. It is able to achieve this set point change into a highly nonlinear region in a settling time of 5 min. The Hammerstein-based NLMPC is able to reach the pH set point only if the simulation was continued for a long period. This was observed for other set point changes as well. One of the reasons for the poor performance of the Hammerstein-based NLMPC could be a poor model in the neighborhood of the set point. In the Hammerstein model the nonlinearity is at the input, and hence the magnitude of input changes plays an important role during the identification process. For example, in Figure 12 the inputs undergo large changes in magnitude in order to meet the desired set point. If the Hammerstein model is unable to capture the nonlinearities associated with these input changes, then the performance of the NLMPC will deteriorate. On the other hand, for Wiener models the nonlinearity is at the output, and therefore the range of magnitude changes in the output affects the goodness of a Wiener model. From Figure 12 we see that the pH changes are not large in magnitude, and the experiment design included changes of this magnitude. Moreover pH is defined as the negative of the natural logarithm of hydrogen ion concentration. This implies that there is a significant nonlinearity at the output. Thus a Wiener structure is naturally suited to capture the nonlinearity of this pH process. The importance of appropriate magnitude changes in the input excitation is illustrated through this result. *A priori* knowledge of the operating conditions of the plant should be taken into account while designing the input sequence for identification.

The disturbance rejection properties of the three controllers are illustrated in Figure 13. Once again Hammerstein- and Wiener-based MPC schemes outperform the linear MPC. The main nonlinearity of the process lies in the pH part and the linear MPC does not do a good job of dealing with it. The integrator in the linear MPC, however, does take the pH back to its steady-state value.

The Wiener-NLMPC was able to meet several set point changes into nonlinear regions that the other controllers could not handle. Figure 14 shows the performance of the Wiener-NLMPC, and Figure 15 shows the constraints being satisfied in the original domain and the equivalent constraints in the latent space. To better illustrate the constraint satisfaction and the geometry, the upper and lower limits on the inputs and the slew rates were reduced for this run:

$$-0.5 < \Delta u_i < 0.5, \quad -5 < u_i < 5; \quad i = 1, 2.$$

Conclusions

A method of handling constraints for NLMPC using Hammerstein and Wiener structures in the PLS framework is pre-

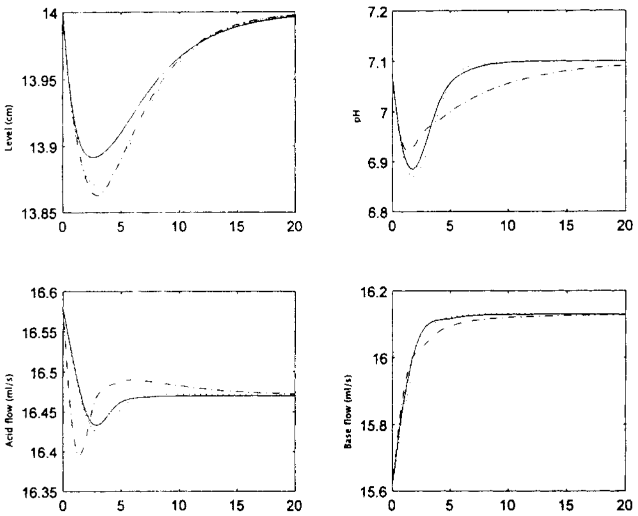

Figure 13. Response of (1) linear MPC (—·), (2) NLMPC-Hammerstein (· · ·), and (3) NLMPC-Wiener (——) to a change in buffer flow (0.6 to 0.2).

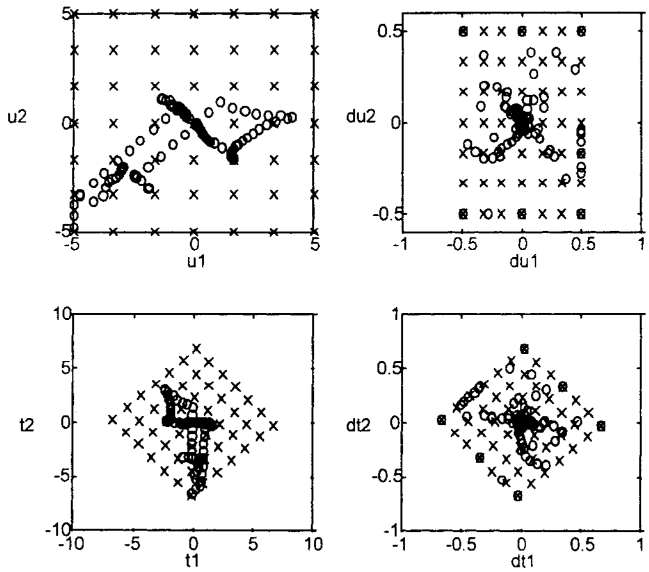

Figure 15. Rate and amplitude constraints in the original space (top graph) and the latent space (bottom graph): × denotes the constraint region, and ○ denotes actual data.

sented in this article. The feasibility of the approach is demonstrated by a simulation case study of an MIMO system involving pH and level control. The main contributions of this work are (1) a technique to effectively use Hammerstein and Wiener structures for modeling and control of MIMO systems in the PLS framework and (2) constrained NLMPC of nonlinear processes using these models. The main advantage of using the PLS framework is that the dynamics are decoupled in the latent space. The constraints, however, get coupled in the latent space. Thus a multivariable approach to controller synthesis has to be adopted for constrained control. For Hammerstein models, the constraints are mapped onto a nonlinear region in the latent space; for Wiener models the constraints, though coupled in terms of the latent variables, maintain their linearity.

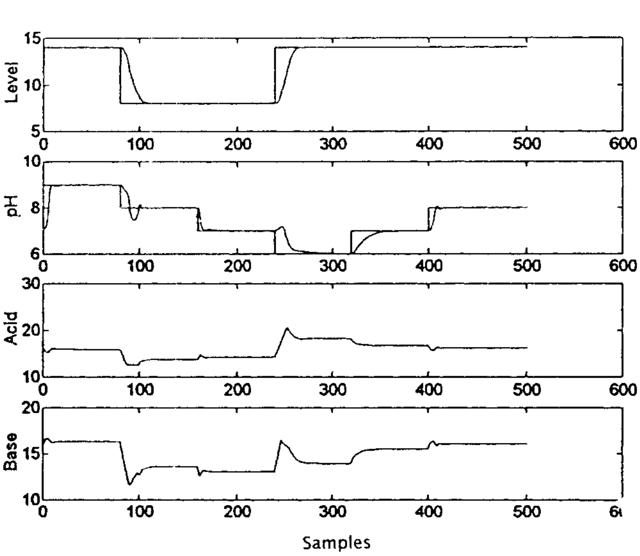

Figure 14. Performance of Wiener-based NLMPC for a series of set point changes.

For Wiener models it is shown that only a quadratic program needs to be solved, which is a significant computational advantage over other NLMPC methods and makes them suitable for controller design purposes. Since block-oriented models serve as useful representations for chemical processes such as distillation and acid-base neutralization (Norquay et al., 1996), it is worthwhile to consider modeling and control of MIMO processes using these structures.

The major advantage of the use of PLS-based Hammerstein and Wiener models is their ability to decouple the dynamics and the nonlinearities. Even though the presence of constraints negates this advantage, it is important to note that the numerical inversion of the static nonlinearities is restricted to a single variable-root finding. In the absence of the PLS structure, the root-finding operation would be a multivariable one and is likely to present numerical difficulties.

Our future research goals include:

- Estimation of uncertainty for PLS-based models.
- Optimal design of experiments for PLS-based nonlinear identification schemes.
- PLS-based modeling and control of nonsquare systems.
- Robust controller design using PLS-based models.

Literature Cited

Bhat, N., and T. J. McAvoy, "Use of Neural Nets for Dynamic Modeling and Control of Chemical Process Systems," *Comput. Chem. Eng.*, **14**, 573 (1993).

Biegler, L. T., "Solution of Dynamic Optimization Problems by Successive Quadratic Programming and Orthogonal Collocation," *Comput. Chem. Eng.*, **8**, 243 (1984).

Brengel, D. D., and W. D. Seider, "Multistep Nonlinear Model Predictive Controller," *Ind. Eng. Chem. Res.*, **28**, 1812 (1989).

de Oliveira, S. L., V. Nevistic, and M. Morari, "Control for Nonlin-

ear Systems Subject to Input Constraints," *Proc. IFAC NOLCOS Symp.*, Lake Tahoe, CA (1995).
Doyle, F. J., III, B. A. Ogunnaike, and R. K. Pearson, "Nonlinear Model-Based Control Using Second Order Volterra Models," *Automatica*, **31**, 697 (1995).
Economou, C. G., and M. Morari, "Internal Model Control 5: Extension to Nonlinear Systems," *Ind. Eng. Chem. Res.*, **25**, 403 (1986).
Eskinat, E., S. H. Johnson, and W. L. Luyben, "Use of Hammerstein Models in Identification of Nonlinear Systems," *AIChE J.*, **37**, 255 (1991).
Garcia, C. E., "Quadratic Dynamic Matrix Control of Nonlinear Processes: An Application to a Batch Reaction Process," AIChE Meeting, San Francisco, CA (1984).
Gattu, G., and E. Zafirou, "Nonlinear Quadratic Dynamic Matrix Control with State Estimation," *Ind. Eng. Chem. Res.*, **31**, 1096 (1992).
Haber, R., and H. Unbehauen, "Structure Identification of Nonlinear Dynamic Systems—A Survey on Input/Output Approaches," *Automatica*, **26**, 651 (1990).
Henson, M. A., and D. E. Seborg, "Adaptive Nonlinear Control of a pH Neutralization Process," *IEEE Trans. Automat. Contr.*, **AC-18**, 552 (1994).
Henson, M. A., and M. J. Kurtz, "Input-Output Linearization of Constrained Nonlinear Processes," AIChE Meeting, San Francisco (1994).
Hernandez, E., and Y. Arkun, "Control of Nonlinear Systems Using Polynomial ARMA Models," *AIChE J.*, **39**, 446 (1993).
Kaspar, M. H., and W. H. Ray, "Chemometric Method for Process Monitoring," *AIChE J.*, **38**, 1593 (1992).
Kaspar, M. H., and W. H. Ray, "Dynamic PLS Modeling for Process Control," *Chem. Eng. Sci.*, **48**, 3447 (1993).
Lakshminarayanan, S., S. L. Shah, and K. Nandakumar, "Identification of Hammerstein Models using Multivariate Statistical Tools," *Chem. Eng. Sci.*, **50**, 3599 (1995).
Lakshminarayanan, S., S. L. Shah, and K. Nandakumar, "Modeling and Control of Multivariable Systems: The Dynamic Projection to Latent Structures Approach," *AIChE J.*, **43**, 2307 (1997a).
Lakshminarayanan, S., R. S. Patwardhan, S. L. Shah, and K. Nandakumar, "A Dynamic PLS Framework for Constrained Model Predictive Control," *Proc. of IFAC Symp. Adv. Control Chem. Processes*, Banff, Alta., Canada, p. 482 (1997b).
Li, W. C., and L. T. Biegler, "Process Control Strategies for Constrained Nonlinear Systems," *Ind. Eng. Res. Chem.*, **27**, 1421 (1988).
Li, W. C., and L. T. Biegler, "Multistep Newton Type Control Strategies for Constrained Nonlinear Processes," *Chem. Eng. Res. Des.*, **67**, 562 (1989).
Manner, B. R., 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**, 1285 (1996).
Mutha, R. K., W. C. Cluett, and A. Penlidis, "Nonlinear Model-Based Predictive Control of Control Nonaffine Systems," *Automatica*, **33**, 907 (1997).
Norquay, S. J., A. Palazoglu, and J. A. Romagnoli, "Nonlinear Model Predictive Control of pH Neutralization using Wiener Models," *Proc. IFAC World Cong.*, San Francisco (1996).
Norquay, S. J., A. Palazoglu, and J. A. Romagnoli, "Unconstrained and Constrained Model Predictive Control using Wiener Models: An Application to pH Neutralization," *Proc. IFAC Symp. Adv. Control Chem. Processes*, Banff, Alta., Canada, p. 541 (1997).
Patwardhan, A. A., J. B. Rawlings, and J. B. Edgar, "Nonlinear Model Predictive Control," *Chem. Eng. Commun.*, **87**, 123 (1990).
Patwardhan, S. C., and K. P. Madhavan, "Nonlinear Model Predictive Control Using Second Order Model Approximation," *Ind. Eng. Chem. Res.*, **32**, 334 (1993).
Qin, S. J., and T. J. McAvoy, "A Data-Based Process Modeling Approach and its Application," *Proc. IFAC Conf. on Dynamics and Control of Chemical Reactors* (DYCORD) (1992).
Ricker, N. L., "The Use of Biased-Least Squares Estimators for Parameters in Discrete Time Impulse Response Models," *Ind. Eng. Chem. Res.*, **27**, 343 (1988).
Verhagen, M., and D. Westwick, "Identifying MIMO Hammerstein Systems in the Context of Subspace Identification Methods," *Int. J. Control*, **63**, 331 (1996).
Wise, B. M., "Adapting Multivariate Analysis for Monitoring and Modeling Dynamic Systems," PhD Thesis, Univ. of Washington, Seattle (1991).
Wold, S., "Estimation of Principal Components and Related Models by Iterative Models Iterative Least Squares," *Multivariate Analysis*, P. R. Krishnaiah, ed., Academic Press, New York, p. 391 (1966).
Zhu, X., and D. E. Seborg, "Nonlinear Predictive Control Based on Hammerstein Models," *Proc. of PSE*, p. 995 (1994).

Manuscript received May 5, 1997, and revision received Jan. 29, 1998.