Modeling and Control of Multivariable Processes: Dynamic PLS Approach

S. Lakshminarayanan, Sirish L. Shah, and K. Nandakumar
Dept. of Chemical Engineering, University of Alberta, Edmonton, Alta., Canada T6G 2G6

The issue of modeling and control of multivariable chemical process systems using the dynamic version of a popular multivariate statistical technique, namely, projection to latent structures (partial least squares or PLS) is addressed. Discrete input-output data are utilized to construct a projection-based dynamic model that captures the dominant features of the process under study. The structure of the resulting model enables the synthesis of a multiloop control system. In addition, the design of feedforward control for multivariable systems using the dynamic PLS framework is also presented. Three case studies are used to illustrate the modeling and control of multivariable linear and nonlinear systems using the suggested approach.

Introduction

Advanced control schemes such as model predictive controllers (DMC, GPC, etc.) are gaining increasingly wide acceptance in the chemical process industries. This is due to their ability to deal with (1) multivariable (square or nonsquare) systems, and (2) systems with hard and soft constraints using simple and intuitive process descriptions such as step/impulse response coefficients and discrete transfer functions. However, the design of such controllers is possible only after the development of a complete model describing the effect of all the process inputs on all the process outputs. First-principles-based models are either difficult to obtain or too unwieldy to use for controller design. The multivariable process model is usually obtained empirically by performing an identification experiment and analyzing the recorded plant input-output data. The presence of several timescales and different delays in multi-input, multioutput (MIMO) processes presents a challenging problem in the identification of such systems. If the system were to exhibit nonlinear characteristics over the desired range of operation, the tasks of identification and control can become even more formidable. Even if an adequate plant model were available, the issue of control structure selection (centralized/decentralized) needs to be addressed. In model-predictive control (MPC), the controller is centralized, and reliability is achieved by performing on-line optimization. Morari (1990) points out that there are many cases where the modeling and design effort necessary for MPC are either impossible or not economically justifiable. In practice, a decentralized (multiloop) control structure is preferred for ease of start-up, bumpless automatic/manual transfer, and fault tolerance in the event of actuator or sensor failures, and is readily designed using recently developed control algorithms (Seborg et al., 1989; Ogunnaike and Ray, 1994) or other methodologies (Morari and Zafiriou, 1989). The decision on loop pairing is critical—the relative gain array (RGA) method and its extensions and physical arguments are the key tools to screen potential alternatives.

Identification of single-input, single-output systems (SISO) is an extremely well-researched topic, even for nonlinear systems (Ljung, 1987; Cinar, 1994). For linear SISO systems, least-squares-based techniques have proven to be handy in the recursive as well as nonrecursive identification schemes. In addition, it is also possible to identify the parameters of all orders, from zero to a user-specified maximum, using an efficient implementation of the least-squares algorithm (Niu and Fisher, 1994). Several commercial identification and control packages (e.g., System Identification Toolbox for use with MATLAB, 1992; ADAPTx, 1992) are capable of estimating linear SISO and/or MIMO dynamic models from observed plant data. In the identification of MIMO processes, a high degree of correlation is often observed between process variables. In such cases, the use of identification software based on the ordinary least-squares technique will result in parameter estimates with large variances owing to the ill-conditioned nature of the problem. One way to circumvent the ill-conditioned nature of the MIMO identification problem is to resort to alternatives other than the ordinary least squares. Very

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

recently, multivariate statistical techniques such as principal components analysis (PCA) and PLS have been applied to chemical engineering problems involving process monitoring, fault detection, and modeling (Kresta, 1992; Wise, 1991; Qin and McAvoy, 1992a; Qin, 1993; Nomikos and MacGregor, 1994; Ricker, 1988). In these applications, the data-compression facility offered by these methods was utilized in condensing the variance of the process into a very low-dimensional latent subspace. This data-compression feature provides a low-dimensional window into the process and facilitates the tasks of monitoring and fault detection (Kresta et al., 1991).

Very few attempts have been made to exploit the potential benefits that PCA and PLS have to offer in the domain of dynamic modeling and control. Ku et al. (1995) have proposed an extension of the standard PCA technique in order to handle dynamic and autocorrelated data. Based on the inclusion of lagged variables in the data matrix, their dynamic PCA (DPCA) method provides monitoring and fault-isolation capabilities under the statistical process control (SPC) framework. Furthermore, DPCA can also identify linear static and dynamic relationships present in data sets, and promises to be an effective tool for process monitoring and identification. Some possible methods (along with their benefits/drawbacks) of using the PLS technique for dynamic model identification are described in Kaspar and Ray (1992, 1993). Qin and McAvoy (1992b) demonstrated the modeling of nonlinear static data using an integrated PLS–neural net structure. The last three articles just cited provided the motivation for the current investigation in which identification and control are performed simultaneously in the PLS framework.

The key contribution of this work involves the development of a modeling approach for MIMO processes that is cast as a series of SISO identification problems. Thus can one, by employing the proposed strategy, utilize the wealth of identification and control algorithms that have been developed for SISO systems. Linear systems are easily handled using standard time series representations (e.g., ARX models); the Hammerstein structure provides a framework for handling nonlinear systems. Although it can be argued that the Hammerstein structure cannot handle every type of nonlinearity, their utility in modeling typical chemical processes (heat exchangers, high-purity distillation columns, acid-base neutralization systems, etc.) has been shown in earlier work (Eskinat et al., 1991; Lakshminarayanan et al., 1995). Subsequent *decentralized* controller design, in the transformed basis set, is based on the estimated SISO dynamic models and therefore provides an automatic selection of loop pairing. The control structure involves the use of pre- and postcompensators along with provisions for annulling the nonlinearities that are identified from the plant data. Finally, we propose a strategy for the design of multivariable feedforward controllers in this PLS framework.

The subject of this article is outlined as follows. First a brief overview of the standard PLS procedure is provided. Extensions of this technique to handle dynamic linear and nonlinear process data in a manner that facilitates easier design of the control system forms the subject of the next two sections. This is followed by a section describing the synthesis of multivariable feedforward controllers. The theoretical matter of this article is supplemented by including several semirealistic examples involving modeling and control of multivariable chemical process systems. Conclusions and proposed future work forms the final section.

Partial Least Squares: Overview

The linear partial least-squares technique has established itself as a robust alternative to the standard least-squares (multiple linear regression) method in the analysis of correlated data. First proposed by Wold (1966), this method has been applied to analyze data in a variety of disciplines, such as the sciences, the social sciences, engineering, and medicine. A tutorial description of PLS, along with a simple example, has been provided by Geladi and Kowalski (1986a,b); for the theoretically inclined reader, Manne (1987) and Höskuldsson (1988) provided excellent analysis of the mathematical properties of the algorithm. In fact, the knowledge and use of PLS has become so commonplace that it warrants no fundamental introduction.

Consider two blocks of measurements X and Y. Often times, we seek to predict the Y space (comprising quality variables), using only the X space (comprising process or causal variables) measurements, with a linear estimator of the form

$$Y = XC + \text{Noise}. \tag{1}$$

Poor performance of the routinely used ordinary least squares (OLS) in the case of correlated measurements makes it necessary to opt for other available choices. Several multivariate techniques, such as canonical correlations analysis (CCA), partial least squares, and principal components regression (PCR), have been proposed for this task. These methods circumvent the collinearity problem associated with multivariate data by constructing and relating *latent* or *virtual* variables (linear combinations of the original variables) instead of the original variables. The philosophy governing the choice of the latent variables for the X space differentiates these methods. Two attributes are of major importance for the estimator: (1) stability, and (2) obtaining a good fit of the data. The linear combinations must account for much of the variation of X and must correlate well with the variables in the Y space to achieve the objectives of model stability and goodness of fit. Each of the multivariate methods accomplishes a different level of balance between these two goals. Wise (1991) describes the nature of these trade-offs in his discussion of continuum regression—a common framework that encompasses all (and even more!) of the multivariate techniques mentioned herein.

Let us assume that the X and Y blocks consist of nx and ny variables, respectively. The number of observations in each of them is N. For practical applications of the PLS algorithm, it may be necessary to scale the X and Y blocks suitably in view of the fact that the measurement units can be grossly different. Without proper scaling, the PLS latent variables may be significantly biased toward variables with larger magnitude. Scaling may be performed using some *a priori* knowledge, for example, assigning larger weights to some key variables; often, all variables are autoscaled (mean centered and scaled to unit variance). This scaling information is stored in the matrices S_x and S_y for the X and Y blocks, respectively. The scaled X and Y blocks, that is, XS_x^{-1} and YS_y^{-1}

are then processed by the PLS algorithm. The raw plant data are assumed to be scaled in this manner in all of the development that follows.

$$S_x = \begin{bmatrix} sx_1 & 0 & 0 & \cdots & 0 \\ 0 & sx_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & sx_{nx} \end{bmatrix} \quad (2)$$

$$S_y = \begin{bmatrix} sy_1 & 0 & 0 & \cdots & 0 \\ 0 & sy_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & sy_{ny} \end{bmatrix}. \quad (3)$$

In PLS, the X and Y data are decomposed as a sum of a series of rank 1 matrices as follows:

$$X = t_1 p_1' + t_2 p_2' + \cdots + t_n p_n' + E = TP' + E \quad (4)$$

$$Y = u_1 q_1' + u_2 q_2' + \cdots + u_n q_n' + F = UQ' + F. \quad (5)$$

In this representation, T and U represent the matrices of scores, while P and Q represent the loading matrices for the X and Y blocks. To determine the dominant directions in which to project data, a maximal description of the covariance within X and Y is used as a criterion. The first set of loading vectors (direction cosines of the dominant directions within the data set), p_1 and q_1, is obtained by maximizing the covariance between X and Y. Projection of the X and Y data, respectively, onto p_1 and q_1 gives the first set of score vectors t_1 and u_1. This procedure is depicted by the block "PLS OUTER MODEL (1)" in Figure 1. The matrices X and Y are now indirectly related through their scores by the "Inner Model," which is just a linear regression of t_1 on u_1 yielding $\hat{u}_1 = t_1 b_1$. $\hat{u}_1 q_1'$ can be interpreted as the part of the Y data that have been predicted by the first PLS dimension; in doing so, the $t_1 p_1'$ portion of X data has been used up. Denoting $E_1 = X$ and $F_1 = Y$, the residuals at this stage are computed by the deflation process (shown as dark squares in Figure 1):

$$E_2 = X - t_1 p_1' = E_1 - t_1 p_1'$$

$$F_2 = Y - \hat{u}_1 q_1' = Y - b_1 t_1 q_1' = F_1 - b_1 t_1 q_1'.$$

The procedure of determining the scores and loading vectors of the inner relation is continued (with the residuals computed at each stage) until the required number of PLS dimensions (n) is extracted. In practice, the number of PLS dimensions is determined based on the percentage of variance explained or by the use of statistically sound approaches such as cross validation. The directions considered irrelevant in the data sets (such as noise and redundancies) are confined to the error matrices E and F.

From a practical viewpoint, PLS can be considered as a technique that breaks up a multivariate regression problem into a series of univariate regression problems. The original

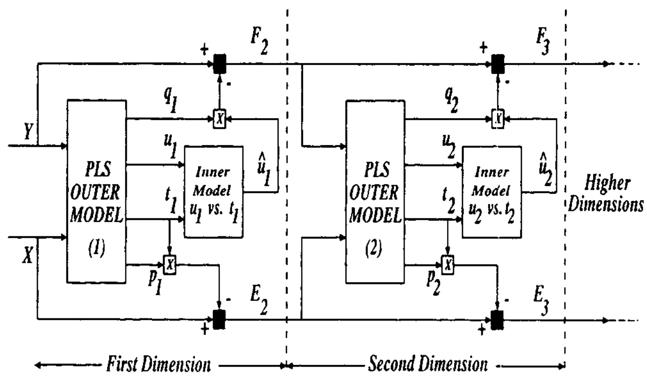

Figure 1. Standard linear PLS algorithm.

regression problem is handled by constructing n inner relationship models (usually, $n \ll nx$). In addition to the PLS outer model (cf. Eqs. 4 and 5), we can write the following equation for describing the inner model of the PLS technique:

$$Y = TBQ' + F \quad (6)$$

In certain versions of the PLS algorithm, the regression coefficients b_i ($i = 1, \ldots, n$) are absorbed into the corresponding q_i vectors. In such cases $b_i = 1 \forall i$; thus B is an identity matrix. The PLS technique has also been cast in the powerful and well-known framework of singular value decomposition (Wise, 1991). It has also been analyzed as an eigenvalue and eigenvector problem (Höskuldsson, 1988) where the mathematical and statistical properties of the PLS algorithm have been enumerated. It has been shown that the latent variables t_i and u_i ($i = 1, \ldots, n$) generated by the PLS algorithm form an orthogonal basis for the X and Y spaces, respectively.

While dealing with nonlinearities in the data, two approaches are possible. The first approach is to include the nonlinear variables (such as squares, exponentials, logarithms) in the appropriate data matrices and use the standard linear PLS procedure described earlier. This would involve dealing with *wider* matrices [for an X matrix with 10 variables, 45 different combinations ($_{10}C_2 = 45$) of two variables can be constructed]. In such circumstances, the higher-order variables tend to dominate the PLS dimensions (Wold et al., 1989), resulting in poor models. An attractive alternative is to move the nonlinearities to the PLS inner model. In the nonlinear PLS algorithm of Wold et al. (1989), the score vectors of the X and Y spaces, that is, t_i and u_i, are related by a quadratic model. It is clearly evident that this strategy can do little when the data are composed of other types of nonlinearities. With their demonstrated utility in approximating arbitrary continuous functions to any desired accuracy, neural networks can be a useful tool in nonparametric modeling studies. To this end, Qin and McAvoy (1992b) proposed an integration of neural networks with the standard PLS algorithm. Their approach preserves the outer relation in linear PLS so as to have the robust prediction property; however, neural networks are employed as the inner regressors. A direct benefit of such a strategy is that only one SISO network is trained at a time. This is not only easier than training a MIMO network, but also circumvents the overparameterization and convergence to local minima problems one usually experiences with a MIMO network. In any case, it is clear that static nonlinearities in data are elegantly handled by incorporating either a parametric (e.g., quadratic polynomial) or a nonparametric (e.g., neural network) regression in the inner relationship of the PLS model.

Dynamic Extension of the PLS Algorithm

The dynamic analog of Eq. 1 can be written as follows:

$$Y = XC_{\text{dyn}} + \text{Noise}, \tag{7}$$

where Y represents the output or controlled variables and X the manipulated variables (inputs). The important difference between Eqs. 1 and 7 is that while C is a matrix of constants in Eq. 1, the matrix C_{dyn} in Eq.7 is a dynamic mapping relating the manipulated inputs to the controlled outputs.

An obvious way to model dynamic processes with PLS is to include past values of the input and/or output variables in the input data matrix X; the algebraic PLS algorithm still forms the computational machinery and does model reduction in a statistically sound manner. This would mean that we need to deal with huge matrices, particularly for MIMO systems. More importantly, the use of the resulting model may only be limited to providing a good input–output mapping of the process rather than aiding the synthesis of a control system (particularly for nonlinear systems). With this approach, C_{dyn} is a matrix of constants whose elements can be interpreted as finite impulse-response (FIR) coefficients (Ricker, 1988) or as a multivariate autoregressive moving average (ARMA) model (Qin and McAvoy, 1992a).

A dynamic PLS modeling procedure that can be directly utilized for control system design has been reported in the literature (Kaspar and Ray, 1992, 1993). Their method does not involve the use of lagged variables, but is based on the filtering of input data. In this way, they argue, the major dynamic component in the data is removed and can be analyzed using the standard PLS procedure. The dynamic filter is designed either by using some *a-priori* knowledge of the process (in the form of an *average* dynamics) or by minimizing the sum of squares of the output residuals, F. In the former case, all the dynamic filters are identical and equal to the assumed *average* dynamics. In the latter case, the dynamic filters are determined using the optimization objective stated earlier, and hence are generally distinct from one another. Using several simulation examples, Kaspar and Ray (1992, 1993) have demonstrated the utility of their approach for the identification and control of systems described by linear models.

We propose a dynamic extension of the PLS algorithm that is based on the direct modification of the PLS inner relation. Instead of relating the input and output scores (i.e., t_i and u_i) using a static linear or nonlinear model, we relate them by a dynamic component such as the ARX or the Hammerstein model. In Kaspar and Ray (1992, 1993), this approach was quickly dismissed as being suboptimal in terms of the PLS outer relationship. This suboptimality problem comes into prominence only when no attention is placed on the design of the plant probing signals. Employing input signals with sufficient low-frequency content, our method identifies *adequate* plant models by utilizing the techniques developed for SISO systems. The proposed strategy will be particularly handy in the modeling of nonlinear multivariable systems—for example, instead of using tedious multivariable Hammerstein models one can piece together several univariate Hammerstein models to obtain an overall model.

For linear systems, although the PLS model matrices and the dynamic inner relationships identified using the proposed strategy and the Kaspar–Ray approach are in general different, it turns out that the mathematical expressions for the steady-state gains, transfer functions, and so forth, are identical. This implies that once the model is identified using either of the methods, it can be used in exactly the same way for the synthesis of feedback and feedforward control structures.

In the PLS algorithm, each of the weight vectors w_i that are used to define the score vectors t_i applies to a different matrix of residuals E_i ($i = 2, \ldots, n$), as

$$t_i = E_i w_i, \tag{8}$$

where w_i is the normalized eigenvector corresponding to the largest eigenvalue of $E_i'F_iF_i'E_i$.

This poses a difficulty in the interpretation of the PLS score vectors, because what is left in the residual matrix E_i at each stage is not clear. For example, some X variables dominate the first few factors and some the later factors. Recognizing this, de Jong (1993) provided the following expression relating the score vectors in terms of the original X matrix:

$$T_{N\times n} = X_{N\times nx} R_{nx\times n}. \tag{9}$$

The matrix R can be expressed in terms of the P and W matrices, as $R = W(P'W)^{-1}$. Note that when all possible PLS components are extracted, the R and P matrices are related as $R^{-1} = P'$. The elements of matrix R will be used to develop the individual transfer functions later.

For the modeling procedure based on incorporation of the linear dynamic relationship (linear systems) in the PLS inner model, the decomposition of the X block is as given by Eq. 4. The dynamic analog of Eq. 5, is given by

$$\begin{aligned} Y &= G_1(t_1)q_1' + G_2(t_2)q_2' + \cdots + G_n(t_n)q_n' + F \\ &= Y_1^{\exp} + Y_2^{\exp} + \cdots + Y_n^{\exp} + F. \end{aligned} \tag{10}$$

Here, the G_i denote the linear dynamic models (e.g., ARX) identified at each state, and $G_i(t_i)q_i'$ quantifies the measure of Y space explained by the ith PLS dimension ($Y_i^{\exp}$). We now define the operator G as the diagonal matrix, comprising the dynamic elements identified at each of the n PLS dimensions, that is,

$$G = \begin{bmatrix} G_1 & 0 & 0 & \cdots & 0 \\ 0 & G_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & G_n \end{bmatrix} \tag{11}$$

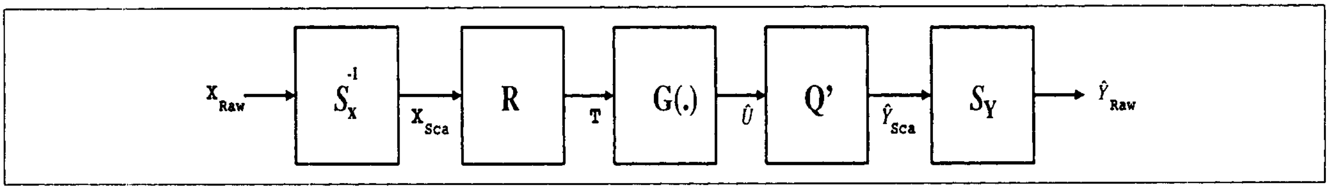

Figure 2. Information flow for the proposed modeling strategy.

A sketch of the dynamic PLS modeling procedure is provided in Figure 2. Note that the scaling information has been explicitly included, using matrices S_x^{-1} and S_y, with the subscript *Sca* denoting the scaled plant data; and $\hat{Y}_{\text{Raw}}$ indicates the predicted values of the controlled outputs.

For the model identified using the dynamic PLS algorithm, we can express the transfer function relating input j to output i as

$$\frac{\Delta y_i(z)}{\Delta x_j(z)} = \frac{sy_i}{sx_j}\left[\sum_{k=1}^{n} R_{jk} G_k(z) Q_{ik}\right], \quad (12)$$

with R_{jk} and Q_{ik} denoting the usual elements of matrices R and Q, respectively. It is seen that the transfer functions relating each plant input to each output is a linear combination of the dynamic elements identified at each PLS dimension. Depending on the relative magnitudes of R_{jk} and Q_{ik}, a particular dynamic component $G_k(z)$ *may or may not* contribute to the overall dynamics of that channel. Equation 12 is useful if it is desired to design a conventional control system for the process.

A frequent argument for employing the PLS technique is its ability to reduce the dimensionality of the measurement space by removing the redundancies present in data sets. This feature of PLS is not utilized in the proposed modeling scheme, owing to the fact that in empirical model identification all of the input information is used in model building. In a real-world situation, it may be possible to perform dimensionality reduction using PLS. If this happens, issues related to the uniqueness of the mapping from the reduced dimensional latent space to the higher dimensional original space would have to be resolved, taking into account the economic value of the concerned manipulated variables.

As already mentioned, we will employ the Hammerstein structure to model nonlinear systems. The Hammerstein model (see Figure 3) consists of a nonlinear static element followed by a linear dynamic element. Hammerstein models are the most simple and useful representations of typical nonlinear chemical engineering processes such as distillation columns, heat exchangers, and pH systems (Eskinat et al., 1991; Luyben and Eskinat, 1994). Well-established linear controller design methods can be employed once the Hammerstein model of the system becomes available. Zhu and Seborg (1994) present the nonlinear predictive control of a neutralization system using a modified Hammerstein representation of the process.

A recent review of the algorithms developed for the identification of the elements in the Hammerstein structure can be found in Lakshminarayanan et al. (1995). By far, the iterative algorithm proposed by Narendra and Gallman (1966) seems to be the most reliable. In the Narendra–Gallman algorithm (NGA), the Hammerstein structure is identified by updating the parameters of the linear dynamic part and the nonlinear static part separately and sequentially. Lakshminarayanan et al. (1995) combined a powerful linear system identification technique, namely canonical variate analysis (Larimore, 1990), and the Narendra–Gallman algorithm to put forth a multivariate Hammerstein-model identification technique.

To model nonlinear systems using the current approach, we relate the score vectors obtained at each PLS dimension using an SISO Hammerstein model ("inner models" in Figure 1). As shown in Figure 3, the score vector t_i is transformed by a nonlinear static relationship (a polynomial of reasonable order) to t_i^*. A linear dynamic model (e.g., ARX model) is then determined between t_i^* and u_i. Although any method can be used for the identification of the Hammerstein models, we use the SISO version of the algorithm presented in Lakshminarayanan et al. (1995). Denoting the identified Hammerstein models by H_i ($i = 1, \ldots, n$), we obtain the equivalent of Eq. 10 as

$$\begin{aligned} Y &= H_1(t_1)q_1' + H_2(t_2)q_2' + \cdots + H_n(t_n)q_n' + F \\ &= Y_1^{\text{exp}} + Y_2^{\text{exp}} + \cdots + Y_n^{\text{exp}} + F. \quad (13) \end{aligned}$$

It must be borne in mind that the Hammerstein structure cannot model every type of nonlinearity. The Hammerstein structure is useful in situations where the process gain changes with the operating conditions, while the dynamics remain fairly constant. When both the process gain and dynamics change significantly over the region of plant operation, it may be necessary to employ *richer* nonlinear models, such as the nonlinear time series models [e.g., nonlinear autoregressive with exogeneous inputs (NARX) and nonlinear autoregressive moving average with exogeneous inputs (NARMAX)] into the PLS inner relationship. The interested reader is referred to Haber and Unbehauen (1990) or Pearson (1994) for practical guidelines on the choice of specific nonlinear dynamic structures.

Illustrative Examples of the Modeling Strategy

The proposed modeling strategy has been applied to several multivariable systems. Three case studies involving distillation columns and an acid–base neutralization system are presented here.

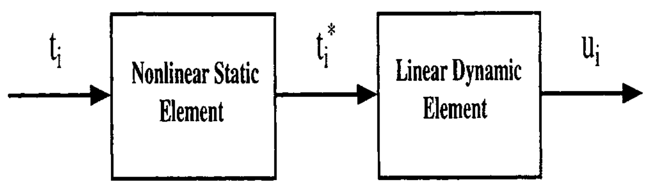

Figure 3. Hammerstein model.

Example 1: distillation column

Wood and Berry (1973) reported the following transfer functions for methanol–water separation in a distillation column. The composition of the top and bottom products expressed in weight % of methanol are the controlled variables. The reflux and the reboiler steam flow rates are the manipulated inputs expressed in lb/min. Time is in minutes.

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{pmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21s+1} \\ \dfrac{6.6e^{-7s}}{10.9s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{pmatrix} \begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix}. \quad (14)$$

The transfer function form of the disturbance channel (feed flow rate and feed composition are the disturbances) is given by

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{pmatrix} \dfrac{3.8e^{-8.1s}}{14.9s+1} & \dfrac{0.22e^{-7.7s}}{22.8s+1} \\ \dfrac{4.9e^{-3.4s}}{13.2s+1} & \dfrac{0.14e^{-9.2s}}{12.1s+1} \end{pmatrix} \begin{bmatrix} d_1(s) \\ d_2(s) \end{bmatrix}. \quad (15)$$

To model the relationship between the manipulated inputs and the controlled outputs, plant data were *collected* by exciting the plant with a series of step changes to the reflux and reboiler steam flow rates. The signal-to-noise ratio (SNR) was set at 10 by adding measurement noise. Following autoscaling of the inputs and outputs, the PLS-based modeling was attempted. The dynamic elements were restricted to be second order (in both numerator and denominator), with delay for both the PLS dimensions. The resulting PLS model is

$$S_x = \begin{bmatrix} 0.4470 & 0 \\ 0 & 0.6374 \end{bmatrix}; \quad S_y = \begin{bmatrix} 12.7577 & 0 \\ 0 & 12.2821 \end{bmatrix}$$

$$P = \begin{bmatrix} 0.3228 & 0.9455 \\ -0.9465 & 0.3256 \end{bmatrix}; \quad R = \begin{bmatrix} 0.3256 & 0.9465 \\ -0.9455 & 0.3228 \end{bmatrix};$$

$$Q = \begin{bmatrix} 0.6972 & 0.7597 \\ 0.7169 & -0.6503 \end{bmatrix}$$

$$G_1 = \frac{0.1417z^{-5}}{1-0.4305z^{-1}-0.4706z^{-2}}$$

$$G_2 = \frac{0.0529z^{-5}+0.0291z^{-6}}{1-0.2336z^{-1}-0.2321z^{-2}}.$$

Figure 4 shows the fit obtained to the plant data with the identified model. Using a different set of plant input–output data, we perform a cross validation test of the identified model. The results, shown in Figure 5, indicate that the identified model provides a good representation of the plant behavior.

Using Eq. 12, the steady-state gain matrix of the identified model is

$$K = \begin{bmatrix} 12.4327 & -18.1543 \\ 6.5938 & -19.3396 \end{bmatrix}.$$

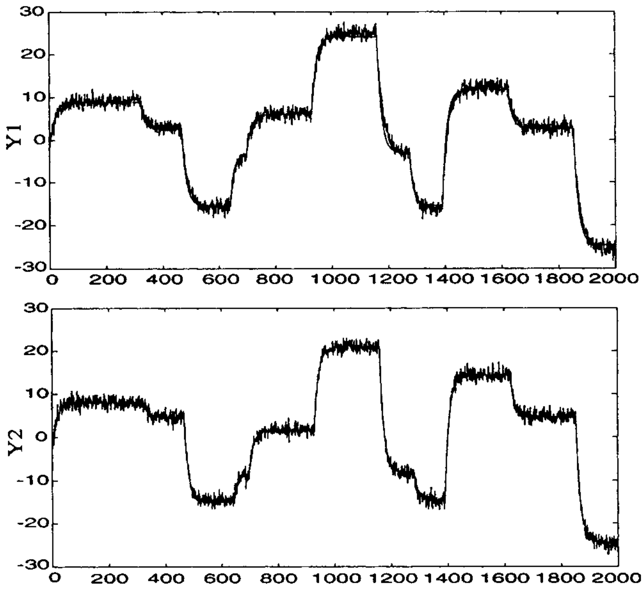

Figure 4. Identification of the Wood-Berry column: model (dashed line) and actual plant (solid line) responses.

This compares quite favorably with the steady-state gains given in Eq. 14.

Example 2: acid–base neutralization process

In many chemical and biochemical processes, control of pH at specified levels is a key requirement. First-principles-based modeling of such systems results in highly nonlinear equations, often with unavailable parameters such as the equilibrium constants. Here, we try to model an acid–base neutralization process using a PLS-based empirical model. The system description, the nonlinear process model, and the operating conditions can be found in Henson and Seborg (1994).

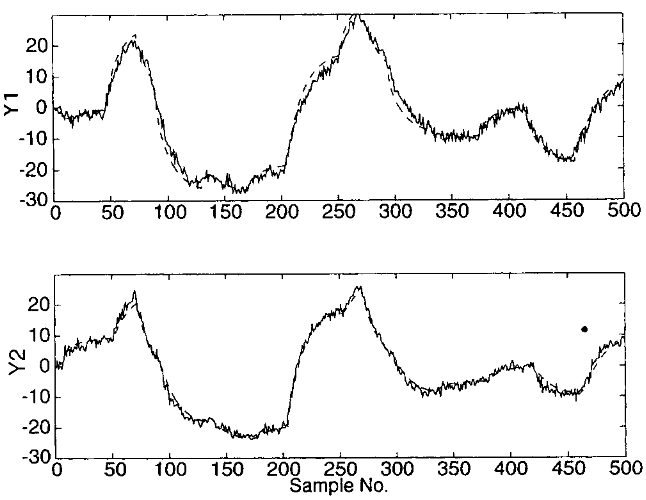

Figure 5. Cross validation for Wood-Berry column: model (dashed line) and actual plant (solid line) responses.

The level and pH of the liquid in the well-stirred neutralization tank are the two outputs that are manipulated by the acid and base flow rates. The nominal value for the level and pH are 14 cm and 7.06, respectively, the acid and base flows being 16.6 mL/s and 15.6 mL/s, respectively. Data were collected by perturbing the system inputs by $\pm 10\%$ of their nominal values using specially designed random signals (Hernández and Arkun, 1993) that enable good nonlinear identification. The signal-to-noise ratio was kept at 10 for identification purposes. The sampling period is 15 s.

The input–output data were first autoscaled and then analyzed using the dynamic PLS algorithm. As a first step, a dynamic model incorporating only linear elements was identified (details are not provided here). As expected, this model was not adequate in modeling the pH of the system. To model the nonlinearities in the system, a Hammerstein model was employed [using the SISO version of the algorithm presented in Lakshminarayanan et al. (1995)] to relate the input and output scores at each dimension. The identified dynamic PLS model is given below:

$$S_x = \begin{bmatrix} 1.1535 & 0 \\ 0 & 1.0122 \end{bmatrix} \tag{16}$$

$$S_y = \begin{bmatrix} 1.2875 & 0 \\ 0 & 1.3079 \end{bmatrix} \tag{17}$$

$$P = \begin{bmatrix} 0.7220 & 0.6796 \\ -0.6919 & 0.7335 \end{bmatrix} \tag{18}$$

$$R = \begin{bmatrix} 0.7336 & 0.6920 \\ -0.6797 & 0.7221 \end{bmatrix} \tag{19}$$

$$Q = \begin{bmatrix} 0.0347 & 1.0000 \\ -0.9994 & 0.0081 \end{bmatrix}. \tag{20}$$

A closer look at the elements of Q indicate that the first PLS dimension essentially models the pH (output variable 2) of the system, while the second PLS dimension models the level (output variable 1). This implies that the nonlinearities are confined to the first PLS dimension, so a Hammerstein model will be needed here. The second dimension can be modeled using only a linear dynamic element.

For the first PLS dimension, a Hammerstein model with *at least* a fourth-order static polynomial was found necessary. However, for control purposes (as we shall see later), it is important that the order of the polynomial be odd, and hence a fifth-order polynomial was chosen to capture the static nonlinearities in the system.

The static nonlinearity (omitting the time variable) is given by

$$t_1^* = 0.02 t_1^5 + 0.1227 t_1^4 - 0.0978 t_1^3 - 0.5909 t_1^2 + t_1. \tag{21}$$

The linear dynamic elements corresponding to the first and second dimensions are

$$G_1 = \frac{0.1617 z^{-1} - 0.0180 z^{-2}}{1 - 0.8849 z^{-1} + 0.0388 z^{-2}} \tag{22}$$

$$G_2 = \frac{0.0458 z^{-1} + 0.0522 z^{-2}}{1 - 0.8744 z^{-1} - 0.0601 z^{-2}} \tag{23}$$

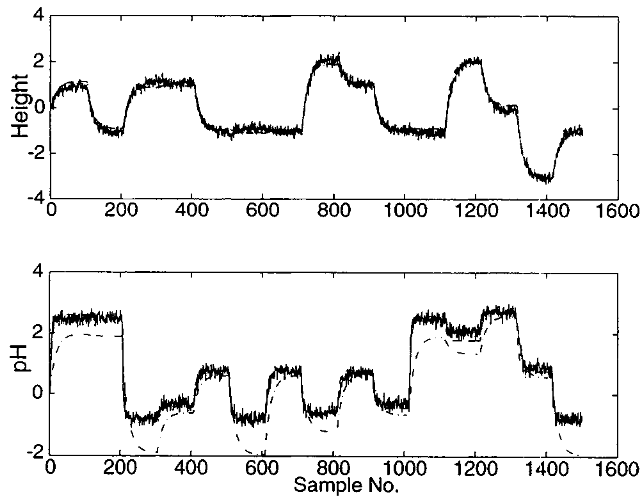

Figure 6. Model fit for the acid–base neutralization system: PLS-Hammerstein model (dashed line), linear model (dashed–dot) and actual plant (solid line).

The results obtained using a Hammerstein model to relate the input and output scores at each dimension are presented in Figure 6. We also show the model fit obtained using only linear inner models. It is observed that the linear model is not able to capture the gain nonlinearities present in the pH measurements. The fit obtained using the Hammerstein inner model is excellent, as is also evident from the cross validation run (Figure 7).

Example 3: multivariable complex distillation column

As a final modeling example, the identification of a 4×4 multivariable distillation column (Alatiqi and Luyben, 1986) is presented. This column (with sidestream stripper) separates a ternary mixture into three products. The four controlled variables include the purities of the three product

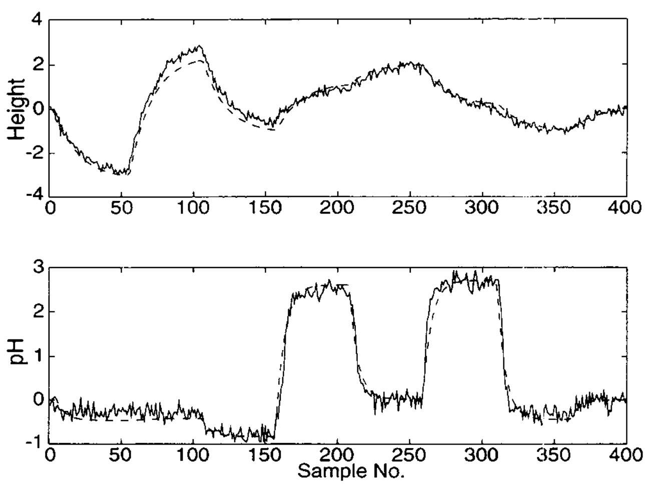

Figure 7. Cross validation with the identified PLS-Hammerstein model for the acid–base neutralization system: model (dashed line) and actual plant (solid line).

streams and a temperature difference between the trays above and below the side draw tray. The reflux flow rate, reboiler heat duty, stripper reboiler heat duty, and the flow rate of feed to the stripper serve as the manipulated variables. The major disturbance to the unit is the feed composition (mole fraction of intermediate component).

Using the transfer-function matrix presented in Alatiqi and Luyben (1986), the plant input–output data were obtained. The plant output data are corrupted both by the load disturbance and measurement noise. In view of space limitations, we do not provide the complete details of the PLS model. The modeling and validation results are summarized in Figures 8 and 9, respectively.

In Figure 8, the loadings plot for the inputs and outputs is illustrated. The uncorrelated nature of the input space and the correlated nature of the output space are depicted in subplots (a) and (b), respectively. It is interesting to see that output variables 2 and 4 are highly correlated. The inner relationship plots between the filtered input scores and the output scores are shown in subplots (c) and (d) for the first two PLS dimensions. The observations fall close to the diagonal on these plots and indicate that the dynamic PLS model will be highly predictive of the outputs. The cross-validation results shown in Figure 9 indicate the good fit provided by the dynamic PLS model, thus confirming the utility of the proposed methodology in the modeling of larger systems.

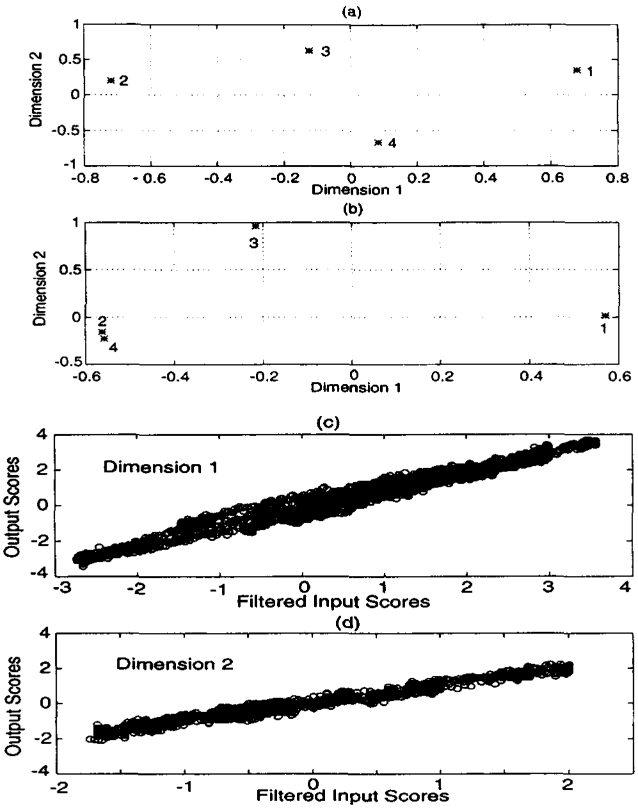

Figure 8. Dynamic PLS modeling strategy for the Alatiqi–Luyben distillation column: (a) loadings plot for the inputs; (b) loadings plot for the outputs; (c) inner relationship plot for dimension 1; and (d) inner relationship plot for dimension 2.

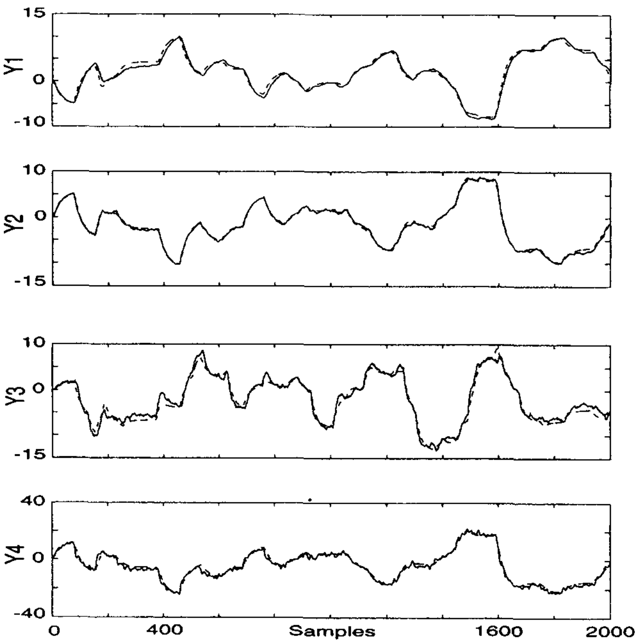

Figure 9. Cross validation for the Alatiqi–Luyben distillation column: model (dashed line) and actual plant (solid line) responses.

Process Control in the PLS Framework

Using linear dynamic PLS models, Kaspar and Ray (1992, 1993) demonstrated a control strategy in which the PLS latent variables (T and U) are directly utilized in the synthesis of the control system. In this approach, the PLS matrices such as S_x, S_y, P, and Q are employed as pre- and postcompensators on the plant. The Q matrix forms a basis for a space into which the scaled output variables are projected, and the P matrix forms a basis onto which the scaled manipulated variables are projected. The controllers are designed independently based on the "inner" dynamic models identified at each dimension. Thus, the controller "sees" the error signals and the command signals in terms of the basis defined by the columns of the respective loading matrices (Q and P). Such a control strategy has a number of advantages. The process is somewhat decoupled owing to the orthogonality of the input scores and the rotation of the input scores to be highly correlated with the output scores. Controller design is simple—any theory available for SISO systems can be used. Because the dynamic part of the PLS model has a diagonal structure, the choice of the input–output pairings is automatic and is optimal in some sense. Infeasible setpoints (in terms of original variables) are not passed on to the controller because only the feasible part of the setpoint vector is retained after it is projected down to the latent variable subspace. This eliminates the problem of multiloop controllers "fighting" each other in a vain bid to reach an impossible setpoint. Due to the nature of the PLS model, nonsquare systems are readily handled.

The Kaspar–Ray scheme is shown in Figure 10. S_x and S_y are the diagonal scaling matrices determined prior to model identification, Q is the loading matrix for the Y block (output variables); Q^{-1} is the appropriate Moore–Penrose inverse of Q; P is the loading matrix for the input (X) block;

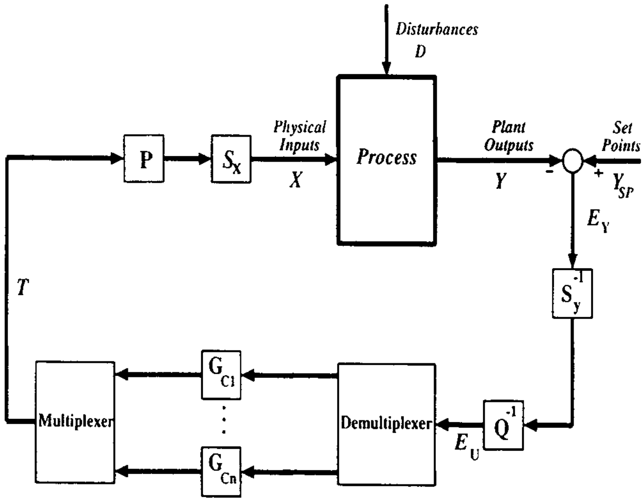

Figure 10. Feedback control using the PLS framework: the Kaspar-Ray scheme for linear systems.

and E_Y is the error in terms of the original output variables. The *projected error* is E_U, upon which the controllers act. The SISO controllers G_{C_1} through G_{C_n} are designed based on the PLS inner models, that is, G_{C_i} is designed based on G_i ($i = 1, \ldots, n$) using any of the available alternatives (e.g., IMC, pole placement, frequency-response techniques). T is the vector of scores computed by the controllers. The scores are then transformed into the real physical inputs that drive the process.

For the physical systems that are modeled by the Hammerstein structure, some modifications to the preceding scheme are necessary. As shown in Figure 11, we now include blocks labeled RF_i ($i = 1, \ldots, n$) after each controller. The controllers are still designed based on the linear dynamic part of the Hammerstein model. Each RF_i is a root-finding routine that is necessary to compensate for the static nonlinear part of the Hammerstein model.

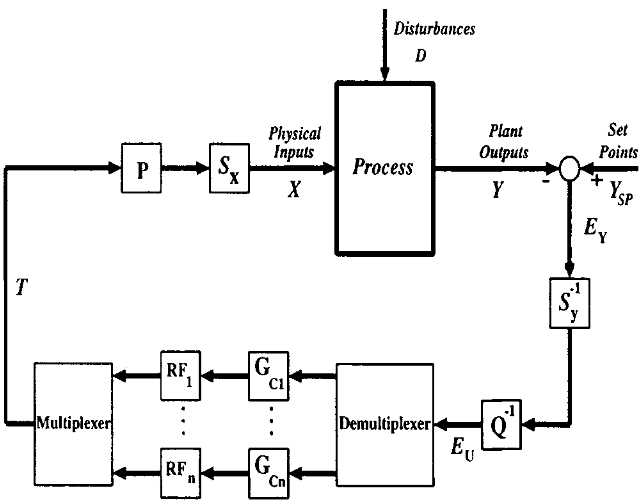

Figure 11. Feedback control using the PLS framework for systems modeled by the Hammerstein structure.

Feedback control of linear systems is well covered by Kaspar and Ray. Here, we shall illustrate the control of the nonlinear acid–base neutralization system using the Hammerstein model that was identified.

Control of the Acid–Base Neutralization Process

In a previous section, we identified a model for the nonlinear pH system. This model will now be used to control the two outputs (level and pH) by manipulating the acid and base flow rates. It is not reasonable to expect a single black-box model (linear or nonlinear) to characterize the steady-state and dynamic features of the process over the entire possible region of plant operation. In general, multiple models are to be identified for the different operating regions, and the control strategy must effectively utilize these models. However, we will employ a single model here and point out its deficiencies in the control of the nonlinear process. Also, there exists a structural mismatch between the identified plant and the real process, which will manifest as a gap between the desired and achieved performance of the control system and may even make the closed-loop system unstable.

The Vogel–Edgar algorithm (Vogel and Edgar, 1980) will be employed in this study to design the controllers G_{C_i}. This algorithm is superior to the minimal prototype controller (in terms of practical applicability) and the Dahlin algorithm (which could lead to the ringing phenomenon), and is ideal for a discrete second-order plus time-delay model. Besides, the Vogel–Edgar algorithm is more robust to modeling errors.

As a first step, we examined whether the linear model identified earlier could be used to control the process. Details of this model were not provided in the section relating to the identification of the pH system, because the closed-loop control system shows sustained oscillations (Figure 12) for a

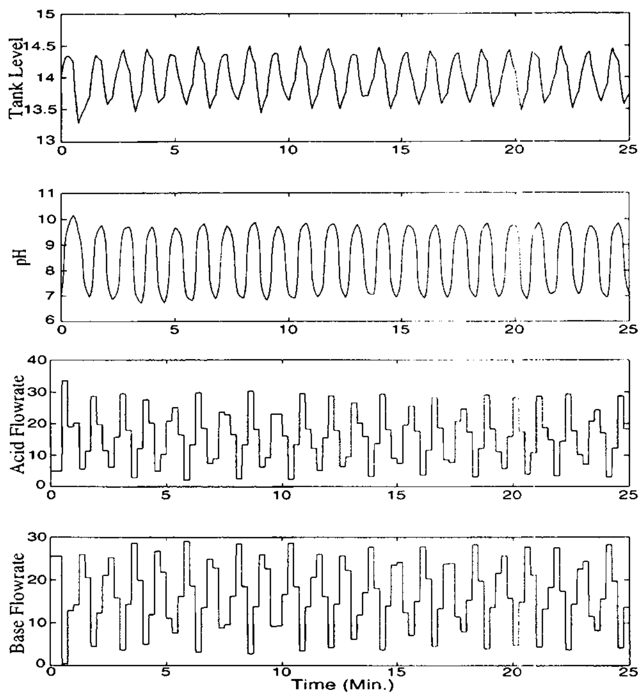

Figure 12. Response to a setpoint change in pH: linear model–linear controller.

setpoint change of + 1.5 in pH (which is within the pH values observed during the identification experiment), thereby establishing the inadequacy of the linear model.

We now utilize the identified nonlinear model as given in Eqs. 16 through 23. The control scheme is as shown in Figure 11, with n (the number of PLS dimensions) equal to 2. G_{C_1} and G_{C_2} are Vogel–Edgar controllers designed based on G_1 (Eq. 22) and G_2 (Eq. 23), respectively. The desired closed-loop settling time is specified as 5 min for both the loops. To compensate for the static nonlinearity observed in the first PLS dimension, the output of controller G_{C_1} is appropriately modified by using the root-finding routine RF_1. The linear nature of the second PLS dimension implies that $RF_2 = 1$ and no modification needs to be performed on the output of G_{C_2}.

Let the output of the controller G_{C_1} be C_1. The output of RF_1 is determined by solving the roots of the polynomial (at each control interval),

$$0.02t_1^5 + 0.1227t_1^4 - 0.0978t_1^3 - 0.5909t_1^2 + t_1 = C_1. \quad (24)$$

Some remarks on the solution of this equation are in order. To ensure that the polynomial has at least one real root, it is necessary that the order of the polynomial be odd. This explains why we employed a fifth-order polynomial even though a fourth-order polynomial gave good fit of the data. Moreover, several real roots may exist—the literature (Anbumani et al., 1981; Bhat et al., 1990) suggest choosing the root with the smallest magnitude.

With the new control strategy in place, the setpoint in pH was changed from 7.06 to 8.5, with the level remaining at 14 cm. The closed-loop response of the system is shown in Figure 13. The required pH value is reached within the desired

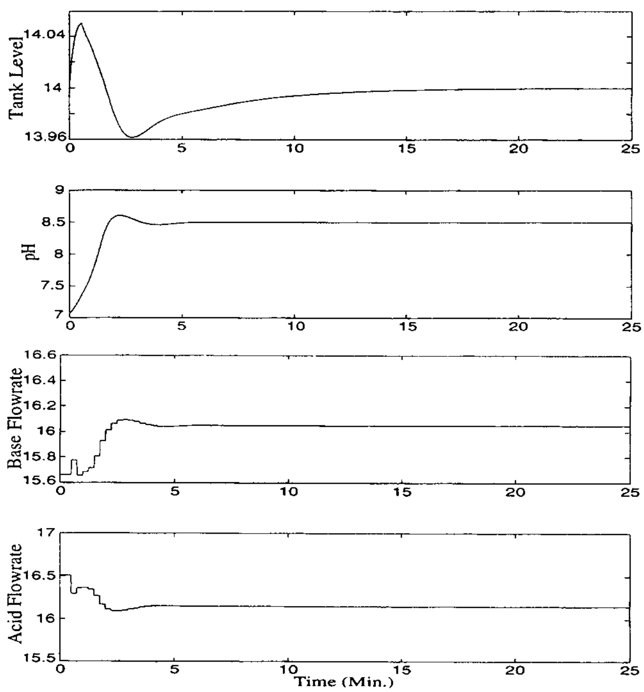

Figure 13. Response to a setpoint change in pH: nonlinear model–nonlinear controller.

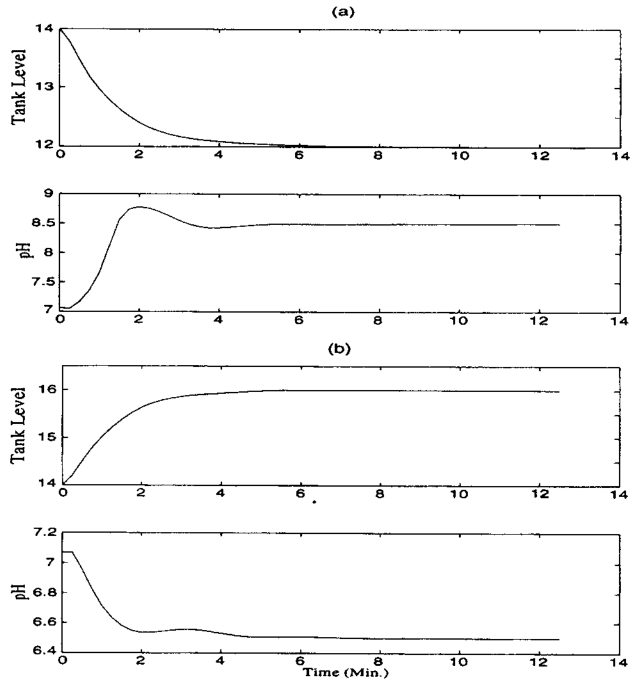

Figure 14. Response to two moderate setpoint changes in level and pH: nonlinear model–nonlinear controller.

time of 5 min. The tank level is only slightly disturbed. The control actions are acceptable and smooth.

Two more runs showing setpoint changes in both the level and pH are presented next. Figure 14 shows the response to changes in setpoint vector from [14 7.06] to (a): [12 8.5], and (b): [16 6.5]. The new setpoints are within the range of values of level and pH used in the identification experiment. The preceding results indicate the utility of the Hammerstein inner model and the workability of the PLS-based nonlinear control strategy.

The closed-loop system is next analyzed for two extreme setpoint changes in level and pH. The new setpoint vectors are [12 10] and [16 5]. In particular, the new pH values are outside the range of values considered in the identification experiment. From the simulation results (Figure 15), it is obvious that the control objectives are not met. This can be attributed to the fact that a single Hammerstein model is inadequate to describe the process behavior over the entire range of operation and highlights the need for a multiple model or an adaptive framework.

The performance of the control system for two unmeasured buffer flow-rate disturbances are shown in Figure 16. In case (a), the buffer flow rate was reduced from its nominal value of 0.6 mL/s to 0.2 mL/s. In case (b), the buffer flow rate was increased from 0.6 mL/s to 1.5 mL/s. Considering that the changes in buffer flow rate lead to large variations in the process gain, the performance of the control scheme is acceptable, though the disturbance can be rejected faster by improved tuning. However, if the buffering content of the system is quite low, the control system exhibits unacceptable oscillatory behavior. An adaptive controller is required to provide better control performance over a wide range of buffering conditions.

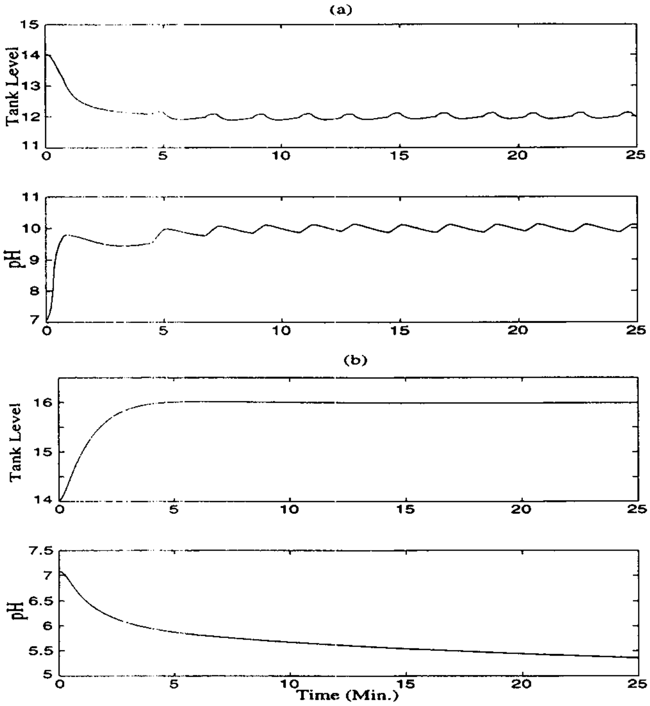

Figure 15. Response to two extreme setpoint changes in level and pH: nonlinear model-nonlinear controller.

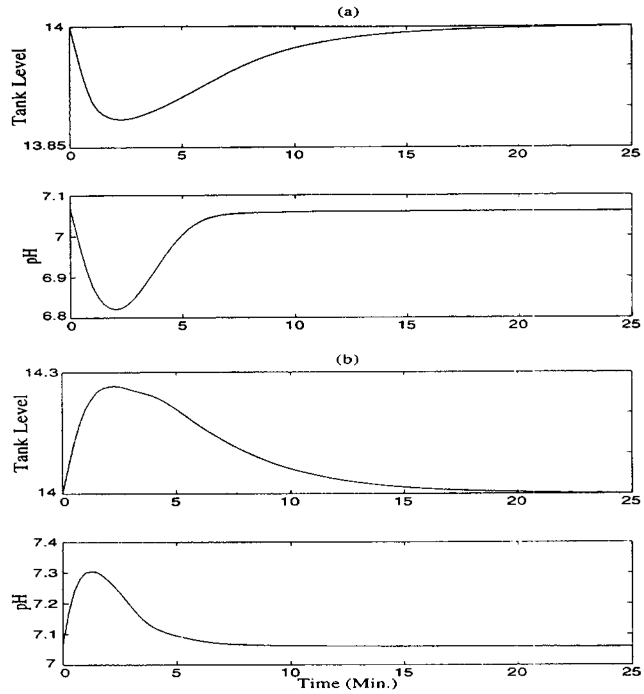

Figure 16. Regulatory response to two step changes in the buffer flow rate: (a) 0.6 mL/s → 0.2 mL/s; (b) 0.6 mL/s → 1.5 mL/s using the nonlinear model and the nonlinear controller.

Multivariable Feedforward Control Using the PLS Framework

One of the primary reasons for the control of industrial processes is to ward off the effects of load disturbances. For disturbances that are *measured*, it is possible to design feedforward controllers that are capable of adjusting the manipulated variables before the controlled variables deviate from their setpoints. Usually feedforward control is never used by itself; it is effective when used in conjunction with feedback control that does not provide satisfactory control performance. Addition of stable feedforward control loops to existing feedback loops on a process does not affect the stability of the closed-loop control system. Furthermore, and more importantly, the performance of the feedforward controller *does not* degrade significantly with modeling errors (Marlin, 1995).

Feedforward controller design for a SISO process depends on the models for the process and disturbance channels. The feedforward controller is the negative of the ratio of the disturbance transfer function to the process transfer function. The SISO feedforward controller is usually realized as a lead/lag element. Often, even the lead/lag and time-delay elements are ignored and a steady-state feedforward controller is employed. The literature is relatively sparse as far as multivariable feedforward controllers are concerned. Shen and Yu (1992) discuss the concept of indirect feedforward control. In their design, fast secondary measurements are used to infer the load changes, and secondary controllers are designed to cancel the effect of these load disturbances on the process outputs. A certain *interaction measure array* (Γ) is defined and is used to design the secondary controllers for quick rejection of specific disturbances. This method is applicable only when some secondary process measurements are available. Stanley et al. (1985) proposed the *relative disturbance gain* (RDG) to compare the disturbance rejection capabilities of the multiloop SISO controllers vs. the inverse-based multivariable controllers such as the decoupler. Shen and Yu (1992) discuss the relationship between Γ and RDG.

Under the assumption that only primary outputs are available for control, we propose a new strategy for the design of a multivariable feedforward controller. Each element of this multivariable controller is realizable as a ratio of two simple transfer functions—this permits retaining the simplicity and elegance of the SISO feedforward design approach. As with SISO feedforward design, the multivariable feedforward controller can be implemented using either lead/lag elements with time delay or just pure gain elements.

First of all, we assume that a dynamic PLS model similar to the one between the manipulated inputs (X) and controlled outputs (Y) is available to describe the relationship between measured disturbances (D) and the controlled outputs (Y). The disturbance-output PLS model (with m PLS dimensions) is characterized by: (1) the diagonal scaling matrices for the output and disturbance spaces—W_y and S_d; (2) the matrices containing the weights attached to the output and disturbance variables in each dimension—Q^d and R^d; and (3) the diagonal matrix G^d containing the dynamic relationship between D and Y—each diagonal element of G^d is denoted by G_j^d ($j = 1, \ldots, m$).

The relationship between the manipulated inputs and the controlled outputs can be summarized by (see Figure 2):

$$Y_x = S_y Q G \{ X S_x^{-1} R \}'. \tag{25}$$

In a similar manner, the relationship between the measured disturbances and the controlled outputs is given by

$$Y_d = W_y Q^d G^d \{DS_d^{-1} R^d\}'. \tag{26}$$

To offset the effect of the measured disturbances on the controlled outputs, we need to determine the required change in the manipulated inputs. This is done by setting $-Y_d = Y_x$ as follows:

$$-W_y Q^d G^d \{DS_d^{-1} R^d\}' = S_y QG\{XS_x^{-1} R\}'. \tag{27}$$

Recognizing that the scores in the manipulated input space (T) and the disturbance variable space (T^d) are defined by $T = XS_x^{-1}R$ and $T^d = DS_d^{-1}R^d$, respectively, we can rewrite Eq. 27 as

$$-W_y Q^d G^d T^{d'} = S_y QGT'. \tag{28}$$

Since the PLS-based control is based on the scores rather than the original variables, we need to express the input scores (T) in terms of the disturbance scores (T^d). Defining $\Lambda = S_y Q$ and $\Omega^d = W_y Q^d$, we can express the preceding equation as

$$-\Omega^d \begin{bmatrix} G_1^d & 0 & 0 & \cdots & 0 \\ 0 & G_2^d & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & G_m^d \end{bmatrix} T^{d'} = \Lambda \begin{bmatrix} G_1 & 0 & 0 & \cdots & 0 \\ 0 & G_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & G_n \end{bmatrix} T'. \tag{29}$$

A rearrangement of this equation gives

$$T' = -\left\{ \Lambda \begin{bmatrix} G_1 & 0 & 0 & \cdots & 0 \\ 0 & G_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & G_n \end{bmatrix} \right\}^{\dagger} \times \Omega^d \begin{bmatrix} G_1^d & 0 & 0 & \cdots & 0 \\ 0 & G_2^d & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & G_m^d \end{bmatrix} T^{d'}, \tag{30}$$

with a further simplification yielding the design equation for the multivariable feedforward controller (in the latent space) as

$$T' = \underbrace{-\begin{bmatrix} G_1 & 0 & 0 & \cdots & 0 \\ 0 & G_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & G_n \end{bmatrix}^{-1} \Lambda^{\dagger} \Omega^d \begin{bmatrix} G_1^d & 0 & 0 & \cdots & 0 \\ 0 & G_2^d & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & G_m^d \end{bmatrix}}_{\text{Feedforward Controller, FFC}} T^{d'}. \tag{31}$$

Element $[i,j]$ ($i = 1, \ldots, n$; $j = 1, \ldots, m$) of the matrix FFC can be written as:

- Case (a): Q is nonsquare ($ny < n$)

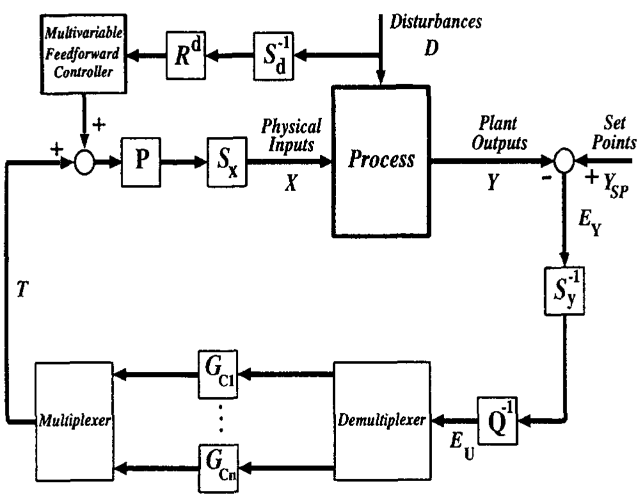

Figure 17. Combined feedback–feedforward control strategy for linear systems: the PLS framework.

$$FFC_{ij} = -\frac{G_j^d}{G_i}\frac{[\det(Q_{ij}^* Q') - \det(Q_i^* Q_i^{*\prime})]}{\det(QQ')}. \quad (32)$$

- Case (b): Q is square (i.e., $ny = n$)

$$FFC_{ij} = -\frac{G_j^d}{G_i}\frac{\det(Q_{ij}^*)}{\det(Q)}. \quad (33)$$

- Case (c): Q is nonsquare ($ny > n$)

$$FFC_{ij} = -\frac{G_j^d}{G_i}\frac{\det(Q'Q_{ij}^*)}{\det(Q'Q)}. \quad (34)$$

In the preceding expressions, the matrix Q_{ij}^* is obtained by replacing the ith column in matrix Q by the weighted jth column of Q^d. Matrix Q_i^* is obtained by simply deleting the ith column in matrix Q:

$$Q_{ij}^* = \left[q_1|q_2|\cdots|q_{i-1}|W_y S_y^{-1} q_j^d|q_{i+1}|\cdots|q_n\right] \quad (35)$$

$$Q_i^* = [q_1|q_2|\cdots|q_{i-1}|q_{i+1}|\cdots|q_n]. \quad (36)$$

It is seen that each element of the multivariable feedforward controller can be expressed as a ratio of two transfer functions multiplied by a constant. This makes the design of a multivariable feedforward control simple and elegant. If only a steady-state feedforward compensation is sought, the dynamic components in Eq. 31 can be replaced with their respective steady-state gains.

Feedforward Control of the Wood–Berry Column

The following PLS model was obtained for the Wood–Berry column by collecting the disturbance–output data, keeping the manipulated inputs stationary:

$$S_d = \begin{bmatrix} 0.2232 & 0 \\ 0 & 1.9753 \end{bmatrix}; \quad W_y = \begin{bmatrix} 1.0012 & 0 \\ 0 & 1.1466 \end{bmatrix}$$

$$P^d = \begin{bmatrix} 0.8233 & 0.4818 \\ 0.5782 & -0.8763 \end{bmatrix}; \quad R^d = \begin{bmatrix} 0.8763 & 0.5782 \\ 0.4818 & -0.8233 \end{bmatrix};$$

$$Q^d = \begin{bmatrix} 0.6949 & 0.2051 \\ 0.7191 & 0.9787 \end{bmatrix}$$

$$G_1^d = \frac{0.0485z^{-6} + 0.0593z^{-7}}{1 - 0.4679z^{-1} - 0.4516z^{-2}}$$

$$G_2^d = \frac{0.0591z^{-6}}{1 - 0.3793z^{-1} - 0.3700z^{-2}}.$$

Using the preceding PLS model and the one obtained earlier (between the manipulated inputs and the controlled outputs), the multivariable feedforward control law is implemented on the Wood–Berry column. The scores for the manipulated inputs computed by the feedforward control law are added to those computed by the feedback controller to obtain a *combined* feedback–feedforward control action (see Figure 17). Regulatory control for two-step disturbances in the feed flow rate (-0.35 units at $t = 0$) and the feed composition (-3 units at $t = 125$ min) are shown in Figure 18.

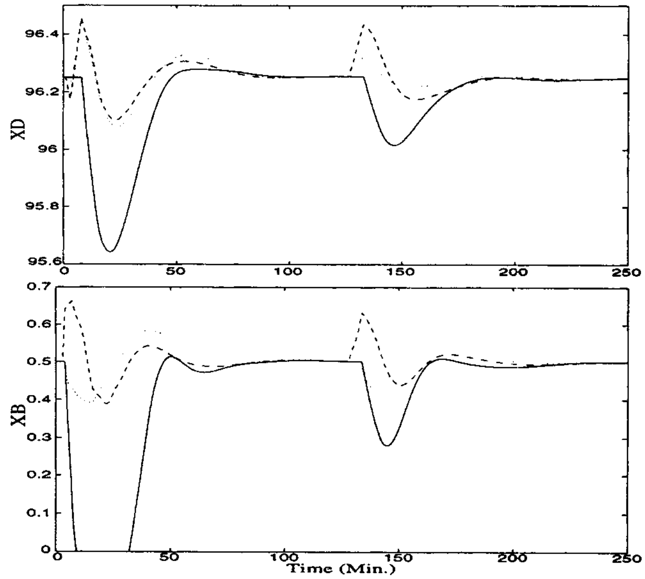

Figure 18. Regulatory control of the Wood–Berry column to a step change of -0.35 units in feed flow rate (at $t = 0$) and a step change of -3 units in feed composition (at $t = 125$ minutes).

Feedback control only (solid line), feedback *plus* steady-state feedforward control (dashed line); and feedback *plus* dynamic feedforward control (dotted line).

From the ISE values reported in Table 1, it is evident that the variations in product quality can be considerably reduced by incorporating feedforward control for these measured disturbances. The ISE values also indicate that, in the presence of model–plant mismatch (as is the case here), a dynamic feedforward controller may not always provide a significantly better performance compared to the more easily implemented steady-state feedforward controller. We noticed significant improvements in control by implementing a single feedforward controller FFC_{11}. This is because the first PLS dimension captures the majority of the variations in the process and disturbance channels.

Feedforward Control of the Acid–Base Neutralization Tank

In this section, we describe the development of the feedforward control strategy for the acid–base neutralization system considered earlier. The buffer flow rate is assumed to be the major measurable disturbance. Instead of employing a modified version of the linear feedforward control law (cf. Eq. 31), we will utilize the structure of the model identified

Table 1. Summary of ISE Values: Wood–Berry Column

Controlled Variable	ISE Values		
	FB Only	FB + Steady-State FF	FB + Dynamic FF
X_D	7.0454	0.8304	0.6791
X_B	7.9183	0.4203	0.3618

for this system (i.e., Eqs. 16 through 23). For this system, it was shown that the first PLS dimension models the pH and the second dimension models the tank level. This implies that once the models relating the buffer flow rate to the pH and level are obtained, two feedforward controllers can be synthesized, one for each dimension.

To obtain the models relating the buffer flow rate to the level and pH, the buffer flow rate was perturbed about its nominal value of 0.6 mL/s by ±0.4 mL/s. During this "experiment," the acid and base flow rates were regulated at their nominal value of 16.6 mL/s and 15.6 mL/s, respectively. These open-loop data were used to construct the models.

The buffer flow-rate vs. pH relationship was modeled by the following Hammerstein model. Denoting the buffer flow rate as d, the static nonlinearity was identified as

$$d^* = 0.0389d^3 - 0.6423d^2 + d. \tag{37}$$

The linear part of the Hammerstein model is

$$G_1^d = \frac{0.1871z^{-1} - 0.0672z^{-2}}{1 - 0.8817z^{-1} + 0.0248z^{-2}}. \tag{38}$$

Note that this linear model relates the transformed buffer flow rate (d^*) to pH. The model fit and the cross validation run using the Hammerstein model are shown in Figure 19—the model appears to capture the relationship to a good measure.

The linear model relating the buffer flow rate to the tank level is

$$G_2^d = \frac{0.1761z^{-1} - 0.1199z^{-2}}{1 - 1.0907z^{-1} + 0.1341z^{-2}}. \tag{39}$$

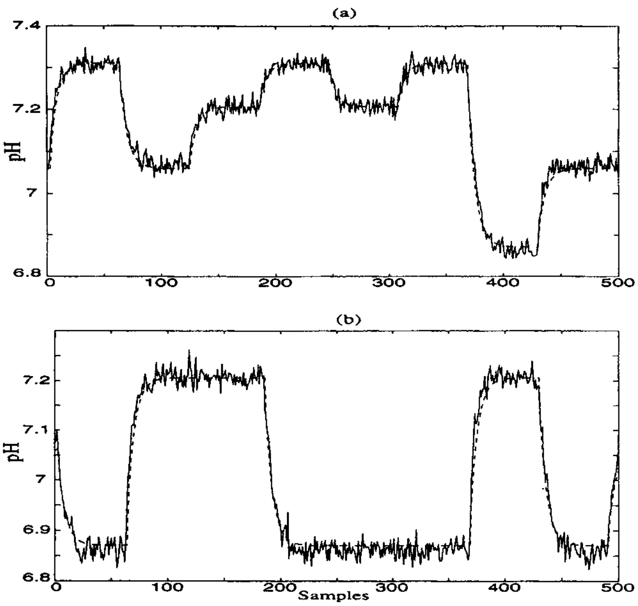

Figure 19. (a) Model fit and (b) cross validation for the buffer flow rate vs. pH relationship: actual pH (solid line) and model predictions (dashed line).

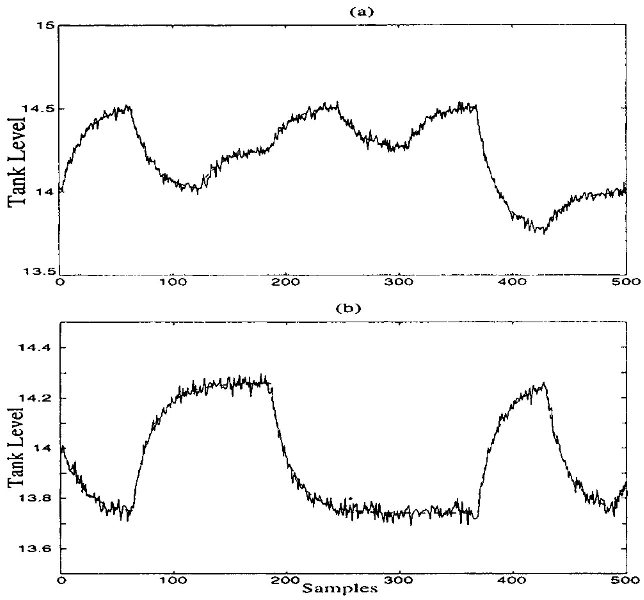

Figure 20. (a) Model fit and (b) cross validation for the buffer flow rate vs. level relationship: actual level (solid line) and model predictions (dashed line).

The validity of the identical model is illustrated by the model fit and cross-validation results shown in Figure 20.

Having obtained convincing models, we are in a position to develop the feedforward controllers.

- *Feedforward Controller for Buffer Flow Rate: Level Subsystem.* Using the information flow diagram (Figure 2), the PLS model (Eqs. 16 through 23), Eq. 39 and the fact that the second PLS dimension essentially models the tank level, we can write

$$Y_1 = Q_{12} sy_1 G_2 t_2 + G_2^d d, \tag{40}$$

with Q_{12} and sy_1 denoting the usual elements in the Q and S_Y matrices, respectively. Setting the deviation variable Y_1 equal to zero in the preceding expression, we obtain the feedforward controller as

$$t_2 = -\left(\frac{1}{Q_{12} sy_1}\right) \frac{G_2^d}{G_2} d. \tag{41}$$

The poles of this feedforward controller were found to lie outside the unit circle, which results in an unstable closed loop. The implementation of this controller was therefore restricted to a steady-state design. The feedforward control action computed by Eq. 41 is superimposed on the output of the feedback controller G_{C_2} shown in Figure 11.

- *Feedforward Controller for Buffer Flow-Rate: pH Subsystem.* The development of this feedforward controller is similar to that presented earlier. Now, we need to look at the first PLS dimension of the model identified in Eqs. 16 through 23. Using Eqs. 37 and 38, we obtain

$$Y_2 = Q_{21} sy_2 G_1 t_1^* + G_1^d d^*, \tag{42}$$

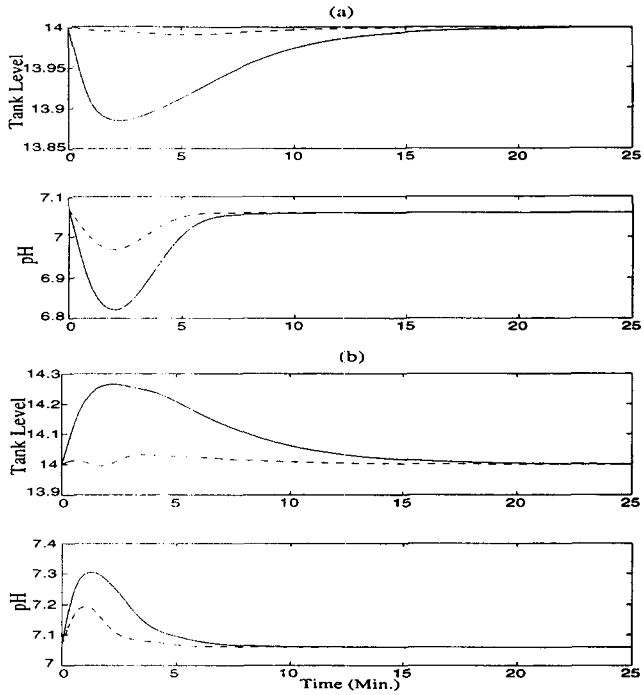

Figure 21. Regulatory response to two step changes in the buffer flow rate: (a) 0.6 mL/s → 0.2 mL/s; (b) 0.6 mL/s → 1.5 mL/s using feedback control only (solid line) and a combined feedback–feedforward control strategy (dashed line).

giving the following feedforward controller

$$t_1^* = -\left(\frac{1}{Q_{21} s y_2}\right)\frac{G_1^d}{G_1} d^*. \tag{43}$$

The measured value of the buffer flow rate (d) is first transformed (using Eq. 37) to d^*. The feedforward control action (t_1^*) is computed using Eq. 43, and is added to the output of controller G_{C_1} in Figure 11.

Comparison of the combined feedback–feedforward control strategy with the feedback control strategy for the same type of disturbances considered earlier is presented in Figure 21. The ISE values for the two control strategies are presented in Table 2. The change in the ISE values for level is dramatic: the ISE values for the combined feedback–feedforward control strategy is only about 1% of that obtained with feedback-only control. Due to the model–plant mismatch, improvements obtained in the ISE values for pH was restricted to about 80–90%.

Table 2. Summary of ISE Values: Acid–Base Neutralization System

	ISE Values (Level)		ISE Values (pH)	
	FB Only	FB + FF	FB Only	FB + FF
Step change of −0.4 unit in buffer flow rate	0.2645	0.0020	0.6027	0.0749
Step change of +0.9 unit in buffer flow rate	1.4110	0.0154	0.5141	0.0995

Conclusions

A data-based approach to the identification and control of multivariable systems has been proposed using the PLS technique. The dynamic PLS model is used for controller synthesis in the transformed basis set employing univariate controller design and tuning techniques. A multivariable feedforward control strategy with a simple structure is also synthesized and incorporated in the latent subspace. The results indicate that the approach may be applicable to a broad class of systems including those with nonlinear characteristics.

The control of the nonlinear acid–base neutralization system was relatively simple, since a Hammerstein structure was found to adequately model the PLS inner relationships. Control of systems whose PLS inner relationships are modeled by nonlinear time-series structures such as NARX and NARMAX may be a bit more formidable and is being examined. To utilize the benefits of advanced model-predictive control algorithms, the linear and nonlinear dynamic PLS models have been integrated into the MPC framework. Some interesting results including the issues involved in the mapping of constraints can be found in Lakshminarayanan (1997) and Lakshminarayanan et al. (1997b).

Literature Cited

ADAPTx, Adaptics, Inc., McLean, VA (1992).

Alatiqi, I. M., and W. L. Luyben, "Control of a Complex Sidestream Column/Stripper Distillation Configuration," *Ind. Eng. Chem. Process Des. Dev.*, **25**, 762 (1986).

Anbumani, K., L. M. Patnaik, and I. G. Sharma, "Self-tuning Minimum Variance Control of Nonlinear Systems of the Hammerstein Model," *IEEE Trans. Automat. Contr.*, **AC-26**, 959 (1981).

Bhat, J., M. Chidambaram, and K. P. Madhavan, "Adaptive Feedforward Control of Hammerstein Nonlinear Systems," *Int. J. Control*, **51**, 237 (1990).

Cinar, A., "Nonlinear Time Series Models for Multivariable Dynamic Processes," *INCINC '94*, First International Chemometrics Internet Conference. Proceedings archived at Web Site http://www.emsl.pnl.gov.2080/docs/incinc/homepage.html (1994).

de Jong, S., "SIMPLS: An Alternative Approach to Partial Least Squares Regression," *Chemometrics Intell. Lab. Syst.*, **18**, 251 (1993).

Eskinat, E., S. H. Johnson, and W. L. Luyben, "Use of Hammerstein Models in Identification of Nonlinear Systems," *AIChE J.*, **37**(2), 255 (1991).

Geladi, P., and B. R. Kowalski, "Partial Least-Squares Regression: A Tutorial," *Anal. Chim. Acta*, **185**, 1 (1986a).

Geladi, P., and B. R. Kowalski, "An Example of 2-Block Predictive Partial Least-squares Regression with Simulated Data," *Anal. Chim. Acta*, **185**, 19 (1986b).

Haber, R., and H. Unbehauen, "Structure Identification of Nonlinear Dynamic Systems—A Survey on Input-Output Approaches," *Automatica*, **26**(4), 651 (1990).

Henson, M. A., and D. E. Seborg, "Adaptive Nonlinear Control of a pH Neutralization Process," *IEEE Trans. Control Syst. Tech.*, **CST-2**(3), 169 (1994).

Hernández, E., and Y. Arkun, "Control of Nonlinear Systems using Polynomial ARMA Models," *AIChE J.*, **39**(3), 446 (1993).

Höskuldsson, A., "PLS Regression Methods," *J. Chemometrics*, **2**, 211 (1988).

Kaspar, M. H., and W. H. Ray, "Chemometric Methods for Process Monitoring and High-performance Controller Design," *AIChE J.*, **38**(10), 1593 (1992).

Kaspar, M. H., and W. H. Ray, "Dynamic PLS Modelling for Process Control," *Chem. Eng. Sci.*, **48**(20), 3447 (1993).

Kresta, J., "Applications of Partial Least Squares Regression," PhD Thesis, McMaster Univ., Hamilton, Ont., Canada (1992).

Kresta, J., J. F. MacGregor, and T. E. Marlin, "Multivariate Statistical Monitoring of Process Operating Performance," *Can. J. Chem. Eng.*, **69**(1), 35 (1991).

Ku, W., R. H. Storer, and C. Georgakis, "Disturbance Detection and Isolation by Dynamic Principal Component Analysis," *Chemometrics Intell. Lab. Syst.*, **30**, 179 (1995).

Lakshminarayanan, S., "Process Characterization and Control Using Multivariate Statistical Techniques," PhD Thesis, Univ. of Alberta, Edmonton, Canada (1997).

Lakshminarayanan, S., S. L. Shah, and K. Nandakumar, "Identification of Hammerstein Models using Multivariate Statistical Tools," *Chem. Eng. Sci.*, **50**(22), 3599 (1995).

Lakshminarayanan, S., R. S. Patwardhan, and S. L. Shah, "A Dynamic PLS Framework for Advanced Process Control," IFAC ADCHEM '97 Meeting, Banff, Canada (1997).

Larimore, W. E., "Canonical Variate Analysis in Identification, Filtering and Adaptive Control," *Proc. IEEE Conf. on Decision and Control*, IEEE, Piscataway, NJ (1990).

Ljung, L., *System Identification: Theory for the User*, Prentice Hall, Englewood Cliffs, NJ (1987).

Luyben, W. L., and E. Eskinat, "Nonlinear Auto-tune Identification," *Int. J. Control*, **59**(3), 595 (1994).

Manne, R., "Analysis of Two Partial-Least-Squares Algorithms for Multivariate Calibration," *Chemometrics Intell. Lab. Syst.*, **2**, 283 (1987).

Marlin, T. E., *Process Control: Designing Processes and Control Systems for Dynamic Performance*, McGraw-Hill, New York (1995).

Morari, M., "Process Control Theory: Reflections on the Past and Goals for the Next Decade," *The Second Shell Process Control Workshop: Solutions to the Shell Standard Control Problem*, D. M. Prett, C. E. García, and B. L. Ramaker, eds., Butterworth, Boston (1990).

Morari, M., and E. Zafiriou, *Robust Process Control*, Prentice Hall, Englewood Cliffs, NJ (1989).

Narendra, K. S., and P. G. Gallman, "An Iterative Method for the Identification of Nonlinear Systems using the Hammerstein Model," *IEEE Trans. Automat. Contr.*, **AC-12**, 546 (1966).

Niu, S., and D. G. Fisher, "Simultaneous Structure Identification and Parameter Estimation of Multivariable Systems," *Int. J. Control*, **59**(5), 1127 (1994).

Nomikos, P., and J. F. MacGregor, "Multi-way Partial Least Squares in Monitoring Batch Processes," *INCINC '94*, First International Chemometrics Internet Conference. Proceedings archived at Web Site http://www.emsl.pnl.gov.2080/docs/incinc/homepage.html (1994).

Ogunnaike, B. A., and W. H. Ray, *Process Dynamics, Modeling and Control*, Oxford Univ. Press, New York (1994).

Pearson, R. K., "Nonlinear Input-Output Modeling," *Proc. IFAC ADCHEM'94 Meeting*, Elsevier Science, New York, p. 1 (1994).

Qin, S. J., "Partial Least Squares Regression for Recursive System Identification," *Proc. Conf. on Decision and Control*, IEEE, Piscataway, NJ (1993).

Qin, S. J., and T. J. McAvoy, "A Data-Based Process Modeling Approach and its Applications," *Proc. IFAC Conf. on Dynamics and Control of Chemical Reactors (DYCORD + '92)*, IEEE, Piscataway, NJ (1992a).

Qin, S. J., and T. J. McAvoy, "Nonlinear PLS Modelling using Neural Networks," *Comput. Chem. Eng.*, **16**(4), 379 (1992b).

Ricker, N. L., "The Use of Biased Least-squares Estimators for Parameters in Discrete-Time Pulse Response Models," *Ind. Eng. Chem. Res.*, **27**, 343 (1988).

Seborg, D. E., T. F. Edgar, and D. A. Mellichamp, *Process Dynamics and Control*, Wiley, New York (1989).

Shen, S. H., and C. C. Yu, "Indirect Feedforward Control: Multivariable Systems," *Chem. Eng. Sci.*, **47**(12), 3085 (1992).

Stanley, G., M. Marino-Galarraga, and T. J. McAvoy, "Shortcut Operability Analysis 1. The Relative Disturbance Gain," *Ind. Eng. Chem. Process Des. Dev.*, **24**, 1188 (1985).

System Identification Toolbox for use with MATLAB, The Mathworks, Inc., Natick, MA (1992).

Vogel, E. F., and T. F. Edgar, "A New Dead Time Compensator for Digital Control," *ISA/80 Proceedings*, ISA, Research Triangle Park, NC (1980).

Wise, B. M., "Adapting Multivariate Analysis for Monitoring and Modeling Dynamic Systems," PhD Thesis, Univ. of Washington, Seattle (1991).

Wold, H., "Estimation of Principal Components and Related Models by Iterative Least Squares," *Multivariate Analysis*, R. P. Krishnaiah, ed., Academic Press, New York, p. 391 (1966).

Wold, S., N. Kettaneh-Wold, and B. Skagerberg, "Nonlinear PLS Modelling," *Chemometrics Intell. Lab. Syst.*, **7**, 53 (1989).

Wood, R. K., and M. W. Berry, "Terminal Composition Control of a Binary Distillation Column," *Chem. Eng. Sci.*, **29**, 1808 (1973).

Zhu, X., and D. E. Seborg, "Nonlinear Predictive Control Based on Hammerstein Models," *Proc. PSE*, Elsevier Science, New York, p. 995 (1994).

Manuscript received Aug. 13, 1996, and revision received Feb. 19, 1997.