LF-LPV Input/Output Data-Based Predictive Controller Design for Nonlinear Systems

Kee-Youn Yoo and Hyun-Ku Rhee
School of Chemical Engineering and Institute of Chemical Processes,
Seoul National University, Seoul 151-742, Korea

Most control engineers concentrate on finding a controller given the plant model or identifying a model from the data. There is no doubt that model-based control and system identification are closely related, simply because one depends strongly on the other. In this work a subspace identification algorithm for LF-LPV (linear-fractional linear parameter-varying) models is reformulated from a control point of view. This algorithm is referred to as an input/output data-based predictive control, in which an explicit model of the system to be controlled is not calculated at any point in the algorithm. It allows for the construction of a nonlinear model predictive controller for an unknown nonlinear system directly from a set of its open-loop measurements. As an example of the input/output data-based predictive control, the styrene solution polymerization in a continuous reactor system is considered to prove the superior performance of LF-LPV input/output data-based predictive controller for polymer quality control. This approach gives a new angle for attacking the problem of identifying and controlling nonlinear systems.

Introduction

A good number of modern control methods use a model to design a control system. When such a model is not available, the design of model-based control, thus, requires a stage of model construction. A typical reason for making a model is to use it for control purposes. The philosophy of model-based control is to calculate the control output as a function of the difference between the desired output of the process and the predicted output of the model. The control output is often calculated as the solution of some quadratic control criterion. Model predictive control is an example of such control methods.

Most researchers either concentrate on finding a controller given the plant model or on identifying a model from the data. There is no doubt that model-based control and system identification are closely related, simply because one strongly depends on the other. The first attempt to combine identification and control resulted in the well-known adaptive control methods. In this case, however, the identification algorithm was originally introduced more as an afterthought than as an integral part of the design. Several researchers started to realize that the identification step, when designing a controller, is far too important to be neglected (Lee, 2000). Thereafter, closed-loop identification and identification for control became common in the control community. The main idea is that the quality of the model and, therefore, the performance of the controller can be improved by alternating closed-loop system identification and subsequent controller design. These methods can be classified as "explicit" tuning methods because a parameterized model of the plant to be controlled is explicitly calculated. The system identification step is in fact nothing more than a vehicle for the controller design. Thus, the system identification step would be reformulated by devising a method that allows for the calculation of a controller directly from the input/output data. This subject is exactly what we are aiming for in this work.

The idea of computing an LQG controller directly from the input/output data was first introduced to a technique

Correspondence concerning this article should be addressed to H.-K. Rhee.

called interactive feedback tuning (IFT) by Hjalmarsson and collaborators (1998). This technique is very closely related to the prediction error method (PEM) in system identification. The IFT method directly calculates the controller parameters from the input/output data by minimizing the control criterion similarly to the PEM. The IFT requires the solution of a nonlinear optimization problem that is typically solved in an iterative way. In this work, we explore how the subspace identification method is related to the model-predictive controller design. The difference between the IFT controller design, and the subspace-based controller design parallels the difference between the PEM based on criterion minimization and the subspace identification based on projections. The advantages and disadvantages of these identification methods are transported from the realm of identification to the realm of control design in terms of optimality, parametric vs. nonparametric methods, and local minima vs. uniqueness of the solution (van Overschee and de Moor, 1996).

Industrial control systems are very often designed on the basis of linear models. When a process is stable and working around a small region of a fixed operating point, one can indeed assume that it can be approximated by a linear system. Especially in the chemical industry, however, processes are strongly nonlinear and the operating point can change frequently and considerably. A typical example is the startup procedure of a distillation column or a chemical reactor. Controlling such a system with a linear model-based controller might lead to poor control performance or even instability. A solution could then be achieved by using nonlinear models that can be applied in a larger region around several operating points.

In this work, we introduce a method based on a linear parameter-varying prediction model for designing a nonlinear model-predictive controller directly from the measured input and output data. Linear parameter-varying (LPV) models are interesting, since on the one hand they are much easier to analyze than general nonlinear models, while on the other hand they are more general than linear models. One could conclude by saying that LPV models are a good compromise between the accuracy of general nonlinear models and the mathematical tractability of linear models.

Linear Parameter-Varying Systems

During the last two decades, there has been an increasing interest in developing systematic, theoretically rigorous techniques for analyzing and designing gain-scheduled control systems for nonlinear plants. Despite the past success of gain scheduling in practice as a nonlinear control technique, little has been known theoretically about the subject. Moreover, the design of the actual gain-scheduled control systems is more of an art than a science. During the late 1980s, Shamma and Athans (1991, 1992) introduced LPV systems as a tool for quantifying such heuristic design rules as "the scheduling parameter must vary slowly" and "the scheduling parameter must capture the nonlinearities of the plant." This has stimulated a great deal of research on LPV systems. These works are related by the use of various types of LPV representation. Employing the velocity-based framework, Leith and Leithead (1998) showed that every smooth nonlinear system of the form $\dot{x} = F(x,u),\ y = G(x,u)$ can indeed be transformed into an LPV form. This reformulation is valid for a very general class of nonlinear systems, and it is emphasized that the LPV representation is valid globally without restriction.

In the literature, the term "linear parameter-varying" is widely employed to refer to any system of the form

$$x_{k+1} = A(\delta_k)x_k + B(\delta_k)u_k$$
$$y_k = C(\delta_k)x_k + D(\delta_k)u_k \tag{1}$$

where δ_k is a scheduling parameter belonging to some class Ω, $x \in \mathbb{R}^{n(=r_0)}$ is the state vector, $u \in \mathbb{R}^m$ is the input vector, and $y \in \mathbb{R}^l$ is the output vector. Although superficially similar, it is emphasized that the dynamic characteristics of the system (Eq. 1) are strongly dependent on the class Ω to which the parameters belong. In particular, when δ_k is permitted to depend on the state x_k, the dependence of the system matrices on the state introduces nonlinear feedback absent in the linear time-invariant/time-varying system. Consequently, in some literature, systems where δ_k may depend on the state are referred to as quasi-LPV systems, while the term LPV is referred to as systems in which δ_k is a strictly exogenous time-varying quantity (strictly independent of the state of the system).

Parameter dependence of LPV systems

In LPV systems, it is assumed that the parameter (and, possibly, its rate of variation), although not known *a priori*, is *on-line measurable*. In order to make theoretical problems tractable, it is often necessary to introduce additional assumptions regarding the way in which δ_k enters the state-space matrices. The most common assumptions are as follows:

- *Affine parameter dependence:* The parameter-varying state-space matrices are affine in δ_k, so that

$$A(\delta_k) = A_0 + \sum_{i=1}^{s} \delta_{k,i} A_i \tag{2}$$

and similar expressions exist for $B(\delta_k)$, $C(\delta_k)$, and $D(\delta_k)$.

- *Multivariate polynomial parameter dependence:* The plant model (Eq. 1) has the state-space matrices whose elements are multivariate polynomial functions in $\delta_{k,1}, \delta_{k,2}, \ldots, \delta_{k,s}$ with coefficients in $\mathbb{R}$. In this case, the state-space matrices have a natural formal series expansion of the form

$$A(\delta_k) = A_0 + \sum_{i=1}^{p} m_i A_i \tag{3}$$

where m_i are monic monomials of the form $\delta_{k,1}^{k_1}, \delta_{k,2}^{k_2}, \ldots, \delta_{k,s}^{k_s}$, with $k_1, k_2, \ldots, k_s \in \mathbb{N}$. This form readily extends to a more general case in which m_i can represent a nonlinear basis function in some other collection of time-varying parameters.

- *Multivariate rational parameter dependence:* The state-space matrices are multivariate rational functions of δ_k. The

rational system models, defined as the ratio of two multivariate polynomial expansions, are able to approximate mathematical functions to the same degree of accuracy with a smaller number of terms.

• *Linear-fractional parameter dependence:* For some time-varying parameters $\delta_k \in \mathbb{R}^s$, we assume that the parameter dependence has the form

$$\begin{pmatrix} A(\delta_k) & B(\delta_k) \\ C(\delta_k) & D(\delta_k) \end{pmatrix} = \begin{pmatrix} A_{xx} & B_x \\ C_x & D \end{pmatrix} + \begin{pmatrix} A_{xz} \\ C_z \end{pmatrix} \Delta_k q (I_r - A_{zz}\Delta_k q)^{-1} (A_{zx}\ B_z) \quad (4)$$

where q is a shift operator and

$$\Delta_k = \begin{pmatrix} \delta_{k,1} I_{r_1} & & \\ & \ddots & \\ & & \delta_{k,s} I_{r_s} \end{pmatrix} \in \mathbb{R}^{r \times r},\ r = \sum_{i=1}^{s} r_i \quad (5)$$

and

$$M = \begin{pmatrix} A_{zz} & A_{zx} & B_z \\ A_{xz} & A_{xx} & B_x \\ C_z & C_x & D \end{pmatrix} \quad (6)$$

The system dynamics is therefore equivalent to

$$\begin{pmatrix} x_{k+1} \\ z_{k+1} \\ y_k \end{pmatrix} = \begin{pmatrix} A_{xx} & A_{zx} & B_x \\ A_{xz} & A_{zz} & B_z \\ C_x & C_z & D \end{pmatrix} \begin{pmatrix} x_k \\ w_k \\ u_k \end{pmatrix}$$
$$w_k = \Delta_k z_k \quad (7)$$

Given exact or approximate expressions for the state-space matrices (as multivariate polynomials and/or rational functions), we can represent the plant using a linear-fractional transformation (LFT). The modeling problem reduces to finding suitable block dimensions $r_1, \ldots, r_s$ for Δ_k and constant, compatible dimension matrix M. This problem is identical to a multidimensional system realization problem (Bose, 1982).

In this work, we refer to the LPV systems with the linear fractional parameter dependence as LF-LPV systems, and use the LF-LPV models as a tool for nonlinear system modeling.

An input/output description of LF-LPV systems

Consider a class of systems, each of which can be represented as an LFT on some time-varying block structure $\mathfrak{D}_k = \text{diag}\,(I_n q, \Delta_k q)$, that is

$$y_k = \mathfrak{F}_u(\mathfrak{M}, \mathfrak{D}_k) u_k \quad (8)$$

in which $\mathfrak{F}_u$ is an upper LFT operator and the system realization $\mathfrak{M}$ is partitioned according to the block structure

$$\mathfrak{M} = \begin{pmatrix} A_{00} & A_{01} & \cdots & A_{0s} & B_0 \\ \vdots & \vdots & & \vdots & \vdots \\ A_{s0} & A_{s1} & \cdots & A_{ss} & B_s \\ C_0 & C_1 & \cdots & C_s & D \end{pmatrix} = \begin{pmatrix} \mathfrak{A} & \mathfrak{B} \\ \mathfrak{C} & \mathfrak{D} \end{pmatrix} \quad (9)$$

Using a doubly indexed set of matrices A_{pq}, we define

$$\mathfrak{A}_{\bullet q} = \begin{pmatrix} A_{0q} \\ A_{1q} \\ \vdots \\ A_{sq} \end{pmatrix} \text{ and } \mathfrak{A}_{p\bullet} = \begin{pmatrix} A_{p0} & A_{p1} & \cdots & A_{ps} \end{pmatrix} \quad (10)$$

As in linear systems, it is possible to find an input/output equation relating different data matrices. This equation is fundamental for subspace identification since it permits us to see how the state can be expressed as linear combinations of the past and future input and output. In order to derive an input/output equation, we introduce the $((s+1)^{j-1}(n+r) \times 1)$ vector of state trajectory as

$$x_{q+j-1|q} = \begin{pmatrix} x_{q+j-2|q} \\ \delta_{q+j-1} \otimes x_{q+j-2|q} \end{pmatrix} = \delta_{q+j-1|q} \otimes x_{q|q} \quad (11)$$

where $\otimes$ denotes the Kronecker product and

$$\delta_{q+j-1|q} = \begin{pmatrix} \delta_{q+j-2|q} \\ \delta_{q+j-1} \otimes \delta_{q+j-2|q} \end{pmatrix} \quad (12)$$

with $x_{q|q} = (x_q^T\ w_q^T)^T$ and $\delta_{q|q} = 1$. In what follows we also introduce the $[\sum_{p=0}^{j-1}(s+1)^p m \times 1]$ vector of stacked input as

$$\boldsymbol{u}_{q+j-1|q} = \begin{pmatrix} u_{q+j-1|q} \\ u_{q+j-1|q+1} \\ \vdots \\ u_{q+j-1|q+j-1} \end{pmatrix} \quad (13)$$

where

$$u_{q+j-1|q+\alpha} = \begin{pmatrix} u_{q+j-2|q+\alpha} \\ \delta_{q+j-1} \otimes u_{q+j-2|q+\alpha} \end{pmatrix} = \delta_{q+j-1|q+\alpha} \otimes u_{q+\alpha|q+\alpha} \quad (14)$$

with initial condition $u_{q+\alpha|q+\alpha} = u_{q+\alpha}$. Similarly, define the vector of stacked output $\boldsymbol{y}_{q+j-1|q}$. Based on the previously defined vectors, the state-transition equation can be derived

from the system description (Eq. 7) as follows:

$$\begin{pmatrix} x_{q+1} \\ z_{q+1} \end{pmatrix} = \mathfrak{A}\ \mathrm{diag}\left[I_n, \Delta_q\right]\begin{pmatrix} x_q \\ z_q \end{pmatrix} + \mathfrak{B}u_q$$

$$= \underbrace{\mathfrak{A}}_{\mathfrak{A}^1} x_{q|q} + \underbrace{\mathfrak{P}_1}_{\mathfrak{P}_1} u_{q|q}$$

$$= \mathfrak{A}^1 x_{q|q} + \mathfrak{P}^1 \boldsymbol{u}_{q|q}$$

$$\begin{pmatrix} x_{q+2} \\ z_{q+2} \end{pmatrix} = \mathfrak{A}\ \mathrm{diag}\left[I_n, \Delta_{q+1}\right]\begin{pmatrix} x_{q+1} \\ z_{q+1} \end{pmatrix} + \mathfrak{B}u_{q+1}$$

$$= \mathfrak{A}\ \mathrm{diag}\left[I_n, \Delta_{q+1}\right]\left(\mathfrak{A}^1 x_{q|q} + \mathfrak{P}^1 \boldsymbol{u}_{q|q}\right) + \mathfrak{P}_1 u_{q+1|q+1}$$

$$= \underbrace{\mathfrak{A}\ \mathrm{diag}\left[\mathfrak{A}^1_{0\bullet}, \ldots, \mathfrak{A}^1_{s\bullet}\right]}_{\mathfrak{A}^2} x_{q+1|q} + \underbrace{\begin{pmatrix} \mathfrak{P}_2 & \mathfrak{P}_1 \end{pmatrix}}_{\mathfrak{P}^2}\begin{pmatrix} u_{q+1|q} \\ u_{q+1|q+1} \end{pmatrix}$$

$$= \mathfrak{A}^2 x_{q+1|q} + \mathfrak{P}^2 \boldsymbol{u}_{q+1|q}$$

$$\vdots$$

from which we obtain

$$\begin{pmatrix} x_{q+j} \\ z_{q+j} \end{pmatrix} = \mathfrak{A}^j x_{q+j-1|q} + \mathfrak{P}^j \boldsymbol{u}_{q+j-1|q} \tag{15}$$

where

$$\mathfrak{P}_j = \begin{pmatrix} \mathfrak{P}_{j0} \\ \mathfrak{P}_{j1} \\ \vdots \\ \mathfrak{P}_{js} \end{pmatrix},\ \mathfrak{P}_{j+1} = \begin{pmatrix} \mathfrak{A}_{\bullet 0}\mathfrak{P}_{j0} & \mathfrak{A}_{\bullet 1}\mathfrak{P}_{j1} \cdots \mathfrak{A}_{\bullet s}\ \mathfrak{P}_{js} \end{pmatrix} \tag{16}$$

$$\mathfrak{A}^j = \mathfrak{A}\ \mathrm{diag}\left[\mathfrak{A}^{j-1}_{0\bullet}, \ldots, \mathfrak{A}^{j-1}_{s\bullet}\right] \tag{17}$$

$$\mathfrak{P}^j = \begin{pmatrix} \mathfrak{P}_j \cdots \mathfrak{P}_1 \end{pmatrix} \tag{18}$$

In addition, the state trajectory is related to the input and output data through the following input/output equation

$$\boldsymbol{y}_{q+j-1|q} = \Gamma^j x_{q+j-1|q} + \mathfrak{H}^j \boldsymbol{u}_{q+j-1|q}$$

$$= \Gamma^j \delta_{q+j-1|q} \otimes x_{q|q} + \mathfrak{H}^j \boldsymbol{u}_{q+j-1|q} \tag{19}$$

in which we introduce the block matrices

$$\Gamma^j = \begin{pmatrix} \Gamma^{j-1} & & & \\ & \Gamma^{j-1} & & \\ & & \ddots & \\ & & & \Gamma^{j-1} \\ C_0\mathfrak{A}^{j-1}_{0\bullet} & C_1\mathfrak{A}^{j-1}_{1\bullet} & \cdots & C_s\mathfrak{A}^{j-1}_{s\bullet} \end{pmatrix} \tag{20}$$

$$\mathfrak{H}^j = \begin{pmatrix} \mathfrak{H}^{j-1} & & & & \\ & \mathfrak{H}^{j-1} & & & \\ & & \ddots & & \\ & & & \mathfrak{H}^{j-1} & \\ C_0\mathfrak{P}^{j-1}_{0\bullet} & C_1\mathfrak{P}^{j-1}_{1\bullet} & \cdots & C_s\mathfrak{P}^{j-1}_{s\bullet} & D \end{pmatrix} \tag{21}$$

with $\Gamma^1 = \mathfrak{C}$ and $\mathfrak{H}^1 = D$.

Matrix input/output description of LF-LPV systems

Block Hankel matrices play an important role in subspace identification algorithms. These matrices can be easily constructed from the given input/output data. Suppose we are given measurements of y_k, u_k, and δ_k for $k = 1, 2, \ldots, 2j + N - 1$. For LF-LPV systems, we define the input block Hankel matrices as

$$U_p = \begin{pmatrix} \boldsymbol{u}_{j|1} & \boldsymbol{u}_{j+1|2} & \cdots & \boldsymbol{u}_{j+N-1|N} \end{pmatrix}$$

$$U_f = \begin{pmatrix} \boldsymbol{u}_{2j|j+1} & \boldsymbol{u}_{2j+1|j+2} & \cdots & \boldsymbol{u}_{2j+N-1|j+N} \end{pmatrix} \tag{22}$$

Here the number of block rows, j, is a user-defined index that is large enough. It should at least be larger than the order of the system one wants to identify. Note that U_p consists of $((s+1)^j - 1)$ m/s rows. The output block Hankel matrices Y_p and Y_f are defined in a similar way. State trajectory $X_{j|1}$ is defined as

$$X_{j|1} = \begin{pmatrix} x_{j|1} & x_{j+1|2} & \cdots & x_{j+N|N} \end{pmatrix} \tag{23}$$

Analogously to U_p and Y_p, we divide the state trajectory into two parts, X_p and X_f

$$X_p = X_{j|1} \text{ and } X_f = X_{2j|j+1} \tag{24}$$

Using the input/output Eqs. 15 and 19, the measurements can be modeled by the matrix input/output equations

$$Y_p = \Gamma^j X_p + \mathfrak{H}^j U_p$$

$$Y_f = \Gamma^j X_f + \mathfrak{H}^j U_f$$

$$X_{j+1|j+1} = \mathfrak{A}^j X_p + \mathfrak{P}^j U_p \tag{25}$$

The derivation of the preceding equations is quite straightforward and very similar to the matrix input/output equations for a linear system.

In what follows we use the truncated linear output Hankel matrix

$$Y_f^l = \begin{pmatrix} y_{j+1} & y_{j+2} & \cdots & y_{j+N} \\ y_{j+2} & y_{j+3} & \cdots & y_{j+N+1} \\ \vdots & & \ddots & \\ y_{2j} & y_{2j+1} & \cdots & y_{2j+N+1} \end{pmatrix} \tag{26}$$

and then deliberate over the truncated matrix input/output equation

$$Y_f^l = \check{\Gamma}^j X_f + \check{\mathfrak{C}}^j U_f = \check{\Gamma}^j \Delta_f \odot X_{j+1|j+1} + \check{\mathfrak{C}}^j U_f \quad (27)$$

where

$$\Delta_f = \begin{pmatrix} \delta_{2j|j+1} & \delta_{2j+1|j+2} & \cdots & \delta_{2j+N-1|j+N} \end{pmatrix}$$

and $\odot$ is the Khatri–Rao product defined as the column-wise Kronecker product of two arbitrary matrices $F \in \mathbb{R}^{f \times h}$ and $G \in \mathbb{R}^{g \times h}$

$$F \odot G = (f_1 \otimes g_1 \cdots f_h \otimes g_h)$$

The matrix input/output equations derived in this subsection are particularly interesting, since they allow for the direct calculation of the controller parameters with the input/output data in the next section.

Approximate matrix input/output description of LF-LPV systems

Let us first look for a linear subspace that approximately contains the state sequence X_p. The idea is to replace the unknown state sequence X_p with an estimate $\hat{X}_p$ from a linear optimal state observer. Let

$$\hat{X}_p = K_1 Y_p + K_2 U_p \quad (28)$$

be the linear optimal estimate of X_p. The estimate of future state sequence X_f is, thus, derived as

$$\hat{X}_f = \Delta_f \odot \hat{X}_{j+1|j+1} = \Delta_f \odot \left(\mathfrak{A}^j (K_1 Y_p + K_2 U_p) + \mathfrak{P}^j U_p \right) = \Delta_f \odot \underbrace{\left(\mathfrak{A}^j K_1 \vdots \mathfrak{A}^j K_2 + \mathfrak{P}^j \right)}_{K} \underbrace{\begin{pmatrix} Y_p \\ U_p \end{pmatrix}}_{W_p} \quad (29)$$

The estimation error is then orthogonal to the block Hankel matrices Y_p and U_p. Using Eqs. 27 and 29, the observation can be modeled by the matrix equation

$$Y_f^l \simeq \check{\Gamma}^j \Delta_f \odot K W_p + \check{\mathfrak{C}}^j U_f = L_1 \Delta_f \odot W_p + L_2 U_f \quad (30)$$

Input/Output Data-Based Predictive Control for LF-LPV Models

As an alternative to "full-fledged" nonlinear model-predictive control, we introduce a different model-predictive control algorithm that uses linear-fractional linear parameter-varying (LF-LPV) prediction models originated from the LF-LPV subspace identification method (Yoo, 2000). In this way, the problem setup of the standard nonlinear model-predictive control approach is avoided, the performance analysis may be simplified, and the computational efficiency is significantly improved. When compared with the standard local linearization-based model-predictive control that uses linear time-invariant prediction models, the input/output data-based predictive control using the LF-LPV prediction model provides more accurate approximations to the true nonlinear system, resulting in a performance closer to that of nonlinear model-predictive control.

LF-LPV prediction model along the parameter trajectories

Assume that the parameter pretrajectory, or the so-called nominal trajectory, over the prediction horizon, N_p, given by

$$\delta_f = \begin{pmatrix} \delta_{t+1} & \delta_{t+2} & \cdots & \delta_{t+N_p} \end{pmatrix} \quad (31)$$

can be determined by some means. The actual trajectories can be measured at real time. The LF-LPV prediction model is given by Eq. 1, for which the system matrices are defined by Eq. 4.

For the design of predictive controller, the subspace identification problem of LF-LPV systems can be interpreted as follows: given the past input and output block Hankel matrices, W_p, and the future input block Hankel matrix, U_f, with the measurable scheduling parameters, δ_k, find an optimal prediction of the future output, Y_f^l. If we use an LF-LPV predictor

$$Y_f^l \simeq L_1 \Delta_f \odot W_p + L_2 U_f \quad (32)$$

where

$$Y_f^l = \begin{pmatrix} y_{N_u+1} & y_{N_u+2} & \cdots & y_{N_u+N} \\ y_{N_u+2} & y_{N_u+3} & \cdots & y_{N_u+N+1} \\ \vdots & & & \\ y_{N_u+N_p} & y_{N_u+N_p+1} & \cdots & y_{N_u+N_p+N-1} \end{pmatrix} \quad (33)$$

$$\Delta_f = \begin{pmatrix} \delta_{N_u+N_p|N_u+1} & \delta_{N_u+N_p+1|N_u+2} & \cdots & \delta_{N_u+N_p+N-1|N_u+N} \end{pmatrix} \quad (34)$$

$$W_p = \begin{pmatrix} Y_p \\ U_p \end{pmatrix} = \begin{pmatrix} \boldsymbol{y}_{N_u|1} & \boldsymbol{y}_{N_u+1|2} & \cdots & \boldsymbol{y}_{N_u+N-1|N} \\ \boldsymbol{u}_{N_u|1} & \boldsymbol{u}_{N_u+1|2} & \cdots & \boldsymbol{u}_{N_u+N-1|N} \end{pmatrix} \quad (35)$$

$$U_f = \begin{pmatrix} \boldsymbol{u}_{N_u+N_p|N_u+1} & \boldsymbol{u}_{N_u+N_p+1|N_u+2} & \cdots & \boldsymbol{u}_{N_u+N_p+N-1|N_u+N} \end{pmatrix} \quad (36)$$

the least-square prediction $\hat{Y}_f^l$ of Y_f^l can be found from the following least-square problem

$$\min_{L_1, L_2} \left\| Y_f^l - (L_1 \; L_2) \begin{pmatrix} \Delta_f \odot W_p \\ U_f \end{pmatrix} \right\|_F^2 \quad (37)$$

The solution to this problem is the orthogonal projection of the row space of Y_f^l into the row space spanned by $\Delta_f \odot W_p$

and U_f, that is

$$\hat{Y}_f^l = Y_f^l / \begin{pmatrix} \Delta_f \odot W_p \\ U_f \end{pmatrix} = \underbrace{Y_f^l /_{U_f} \Delta_f \odot W_p}_{L_1 \Delta_f \odot W_p} + \underbrace{Y_f^l /_{\Delta_f \odot W_p} U_f}_{L_2 U_f}, \quad (38)$$

for which the orthogonal and oblique projections are explained in the Appendix.

The numerical implementation of the projection (Eq. 38) can be done in a very efficient way with a QR decomposition

$$\begin{pmatrix} \Delta_f \odot W_p \\ U_f \\ Y_f^l \end{pmatrix} = \begin{pmatrix} R_{11} & & \\ R_{21} & R_{22} & \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \begin{pmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \end{pmatrix} \quad (39)$$

By denoting

$$L = \begin{pmatrix} R_{31} & R_{32} \end{pmatrix} \begin{pmatrix} R_{11} & \\ R_{21} & R_{22} \end{pmatrix}^{\dagger} \quad (40)$$

where the symbol † denotes the pseudoinverse, it is easy to show that Eq. 38 can be rewritten as

$$\hat{Y}_f^l = Y_f^l / \begin{pmatrix} \Delta_f \odot W_p \\ U_f \end{pmatrix} = L \begin{pmatrix} \Delta_f \odot W_p \\ U_f \end{pmatrix} \quad (41)$$

Using the Matlab notation, we have

$$L_1 = L\big(:, 1{:}\,(l+m)(s+1)^{N_p-1}\big((s+1)^{N_u}-1\big)/s\big),$$
$$L_2 = L\big(:, (l+m)(s+1)^{N_p-1}\big((s+1)^{N_u}-1\big)/s+1{:}\,end\big) \quad (42)$$

In the linear case, the parameter vectors, $\delta_{N_u+N_p+\hat{j}-1|N_u+\hat{j}}$ and $\delta_{N_u+\hat{j}-1|\hat{j}}$ $\hat{j}=1, \ldots, j$, are reduced to 1, and then L_1 and L_2 are given by

$$L_1 = L\big(:, 1{:}\,(l+m)N_u\big),$$
$$L_2 = L\big(:, (l+m)N_u+1{:}\,end\big) \quad (43)$$

Two matrices, L_1 and L_2, will play important roles in the next subsection where the link between subspace identification and model-predictive control is made. It will be shown that they are of direct relevance in the design of the model-predictive controller.

The optimal predicted output sequence $\hat{\boldsymbol{y}}_f^l$ can be expressed as

$$\hat{\boldsymbol{y}}_f^l = L_1 \delta_{t+N_p|t+1} \otimes \boldsymbol{w}_p + L_2 \boldsymbol{u}_f \quad (44)$$

where

$$\hat{\boldsymbol{y}}_f^l = \begin{pmatrix} \hat{y}_{t+1} \\ \vdots \\ \hat{y}_{t+N_p} \end{pmatrix}, \boldsymbol{u}_f = \begin{pmatrix} u_{t+N_p|t+1} \\ \vdots \\ u_{t+N_p|t+N_p} \end{pmatrix} \text{ and } \boldsymbol{w}_p = \begin{pmatrix} y_{t|t-N_u+1} \\ \vdots \\ y_{t|t} \\ u_{t|t-N_u+1} \\ \vdots \\ u_{t|t} \end{pmatrix} \quad (45)$$

Input/output data-based predictive controller

Applying the LF-LPV prediction model, the original nonlinear model-predictive control optimization setup can be transformed into the following

$$\min_{u_{t+1}, \ldots, u_{t+N_p}} \sum_{k=1}^{N_p} l\big[y_{t+k}(\delta_{t+k}), u_{t+k}(\delta_{t+k})\big] \quad (46)$$

subject to the LPV model dynamics

$$x_{k+1} = A(\delta_k)x_k + B(\delta_k)u_k$$
$$y_k = C(\delta_k)x_k + D(\delta_k)u_k$$

and also subject to the constraints

$$c\big(y_{t+k}(\delta_{t+k}), u_{t+k}(\delta_{t+k})\big) \le 0,\ k = 1, \ldots, N_p \quad (47)$$

If the performance index is chosen to be quadratic and the constraints are linear, then the preceding optimization can be reduced to a series of quadratic programming (QP) problems in company with the update strategy of the information about future parameter trajectory.

Instead of minimizing the performance index in the classic way, we start from the input/output equation. The future reference trajectory, r_f, and the future input sequence, u_f, are defined as

$$r_f = \begin{pmatrix} r_{t+1} \\ r_{t+2} \\ \vdots \\ r_{t+N_p} \end{pmatrix} \text{ and } u_f = \begin{pmatrix} u_{t+1} \\ u_{t+2} \\ \vdots \\ u_{t+N_p} \end{pmatrix} \quad (48)$$

For the purpose of reducing the computational complexity, it is very common to introduce constraints on the future controls. An often-used approach is to assume that control increments are equal to zero after N_u ($\le N_p$) steps, that is

$$\Delta_{u_{t+k}} = u_{t+k} - u_{t+k-1} = 0,\ k > N_u \quad (49)$$

It is well known that this also has the effect of producing less aggressive controllers. The quantity N_u is usually referred to as the control horizon. Taking into account the fact that the control horizon, N_u, typically is shorter than the prediction horizon, N_p, and that Eq. 49 is required, one can express u_f

as follows

$$u_f = \Lambda \bar{u}_f = \Lambda \begin{pmatrix} u_{t+1} \\ u_{t+2} \\ \vdots \\ u_{t+N_u} \end{pmatrix} \tag{50}$$

where

$$\Lambda = \begin{pmatrix} I_m & 0 & \cdots & 0 \\ 0 & I_m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I_m \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & I_m \end{pmatrix}$$

The sequence of control increments can also be expressed in vector form

$$\Delta \bar{u}_f = \bar{D} \bar{u}_f - \bar{u}_t \tag{51}$$

by introducing the auxiliary quantities

$$\bar{D} = \begin{pmatrix} I_m & 0 & \cdots & 0 \\ -I_m & I_m & & \vdots \\ 0 & \ddots & \ddots & 0 \\ 0 & \cdots & -I_m & I_m \end{pmatrix} \text{ and } \bar{u}_t = \begin{pmatrix} u_t \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Then the optimal control criterion can be written as

$$J = \left(r_f - y_f^l\right)^T \hat{Q}\left(r_f - y_f^l\right) + \bar{u}_f^T \hat{R} \bar{u}_f + \left(\bar{D}\bar{u}_f - \bar{u}_t\right)^T \hat{R}_\Delta \left(\bar{D}\bar{u}_f - \bar{u}_t\right) \tag{52}$$

where

$$\begin{aligned} \hat{Q} &= \text{diag}\,(Q \cdots Q) \in \mathbb{R}^{N_p l \times N_p l} \\ \hat{R} &= \text{diag}\,(R \cdots R) \in \mathbb{R}^{N_u m \times N_u m} \\ \hat{R}_\Delta &= \text{diag}\,(R_\Delta \cdots R_\Delta) \in \mathbb{R}^{N_u m \times N_u m} \end{aligned} \tag{53}$$

and the input sequence $\boldsymbol{u}_f$ for the prediction input/output equation and the input sequence $\bar{u}_f$ for the control optimization have the following relation

$$\boldsymbol{u}_f = \begin{pmatrix} \delta_{t+N_p|t+1} \otimes I_m & & & \\ & \delta_{t+N_p|t+2} \otimes I_m & & \\ & & \ddots & \\ & & & \delta_{t+N_p|t+N_p} \otimes I_m \end{pmatrix} \times \begin{pmatrix} u_{t+1} \\ u_{t+2} \\ \vdots \\ u_{t+N_p} \end{pmatrix} = \hat{\Delta}_{t+N_p|t+1} u_f = \hat{\Delta}_{t+N_p|t+1} \Lambda \bar{u}_f \tag{54}$$

For the unconstrained case, the minimizing control sequence is obtained explicitly by means of the ordinary least-square theory with the information about future parameter trajectory. For the constrained case, the linear constraints (Eq. 47) can be reformulated as

$$\bar{C}\bar{u}_f \le \bar{c} \tag{55}$$

where

$$\bar{C} = \begin{pmatrix} I_{N_u \times m} \\ -I_{N_u \times m} \\ \bar{D} \\ -\bar{D} \end{pmatrix} \text{ and } \bar{c} = \begin{pmatrix} \mathbf{1}_{N_u} \otimes u_{\max} \\ -\mathbf{1}_{N_u} \otimes u_{\min} \\ \mathbf{1}_{N_u} \otimes \Delta u_{\max} + \bar{u}_t \\ -\mathbf{1}_{N_u} \otimes \Delta u_{\min} - \bar{u}_t \end{pmatrix} \tag{56}$$

In the next subsection, we derive a series of QP problems that can be efficiently solved by using standard numerical optimization software with the updated strategy about the information of future parameter trajectory.

Implementation algorithm

The input/output data-based predictive controller can now be implemented by performing the following procedure for the optimization and updating of nominal trajectories at each sampling time t:

(1) Find a nominal future parameter trajectory for time step t

$$\delta_f = \begin{pmatrix} \delta_{t+1} & \delta_{t+2} & \cdots & \delta_{t+N_p} \end{pmatrix}$$

(2) Compute the control moves

$$u_f = \begin{pmatrix} u_{t+1}^T & u_{t+2}^T & \cdots & u_{t+N_u}^T \end{pmatrix}^T$$

by solving the optimization problem of the predictive control setup.

(3) Update the current parameter trajectory on the basis of the updated future process information $\hat{y}_f^l$ and u_f within the prediction horizon. If the future parameter trajectory converges, go to step 4; otherwise go to step 2.

(4) Implement the first control move u_{t+1}.

(5) Define a future nominal parameter trajectory for the next sampling time t' as follows

$$\begin{aligned} \delta_{t'+1} &= \delta_{t+2} \\ \delta_{t'+2} &= \delta_{t+3} \\ &\vdots \\ \delta_{t'+N_p-1} &= \delta_{t+N_p} \\ \delta_{t'+N_p} &= \delta_{t+N_p} \end{aligned} \tag{57}$$

In order to construct the proposed controller, we have to choose the scheduling parameters to capture the nonlinearity of the system. Nonlinear controllers use a variety of nonlinear models to capture the nonlinearity of the system, and the nonlinear models can be constructed by using the input/output data. In general, the input/output data are on-line mea-

surable or estimated by some experimental correlations. One can also use some combinations of the input/output variables as the scheduling parameters. If the input/output variables for the controller design have an almost linear relationship, it is sufficient to employ a linear controller for the control of that system and there is no need for any scheduling parameters. However, when the input/output variables have a severe nonlinear relationship, we must use a nonlinear controller. In such a case we can choose the scheduling parameters among the input/output variables or from combinations of the input/output variables to capture the nonlinearity. The selection of the scheduling parameters is more of an art than a science.

In this section we have presented an algorithm for the calculation of the input/output data-based predictive controller on the basis of the input/output data, only without any explicit plant-model calculation. Although the derivation is based on expressions from the subspace identification theory, the controller implementation bypasses the identification step. Given the input/output data of the system, one can directly derive the controller parameters by the single-step QR factorization.

Examples: Quality Control in a Continuous Styrene Solution Polymerization Reactor

Our goal here is to study the control of polymer quality such as conversion and weight average molecular weight in a continuous styrene solution polymerization reactor to corroborate the performance of the proposed control algorithm.

Plant description

The plant under consideration is a styrene solution polymerization reactor in a continuous operation mode. In order to perform the simulation study, we consider the reactor model in which the styrene solution polymerization occurs. The reaction kinetics is assumed to follow the free-radical polymerization mechanism. The method of moments is adopted to calculate the number average molecular weight (M_n) and the weight average molecular weight (M_w). The dynamic behavior of the continuous reactor is described by the first-principles model in Na and Rhee (2000, 2002), with the physical properties and kinetic parameters reported in their works (Yoo et al., 1999; Na and Rhee, 2000, 2002). The empirical gel effect correlation by Hamer et al. (1981) is included in the reactor model. This polymerization reactor model was developed for the description of an experimental CSTR system for styrene polymerization.

Figure 1 shows the continuous styrene solution polymerization reactor. Reactants are fed to the reactor at the top. The produced polymer and the unreacted reactants are withdrawn at the bottom of the reactor. The input and output variables and the scheduling parameter are chosen as follows:

- *Inputs:* The manipulated inputs are the feed flow rate, q_f, and the jacket inlet temperature, $T_{j,in}$ which are ranged from 0.005 to 0.030 L/min and from 55 to 85°C, respectively.
- *Outputs:* The outputs to be controlled are the monomer conversion X and the weight average molecular weight M_w, which are measured on-line by using appropriate devices (for example, densitometer and viscometer) and correlations. Sampling times for the measurement of conversion and weight average molecular weight are 1 min.
- *Parameter:* The selected scheduling parameter is the jacket inlet temperature, $T_{j,in}$. In the case of the lab-scale polymerization reactor considered, the jacket inlet temperature follows closely the temperature inside the reactor due to

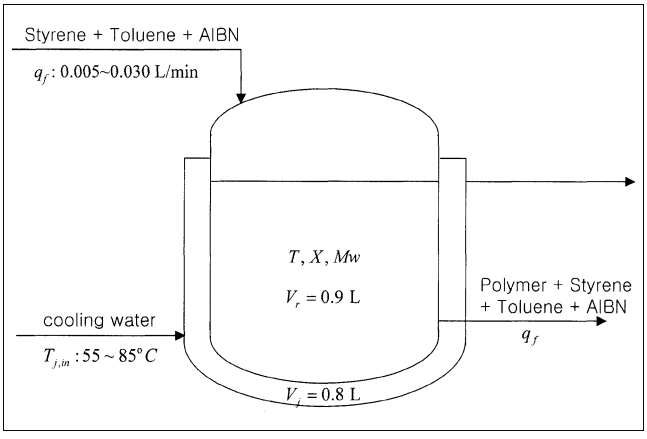

Figure 1. Continuous styrene solution polymerization reactor.

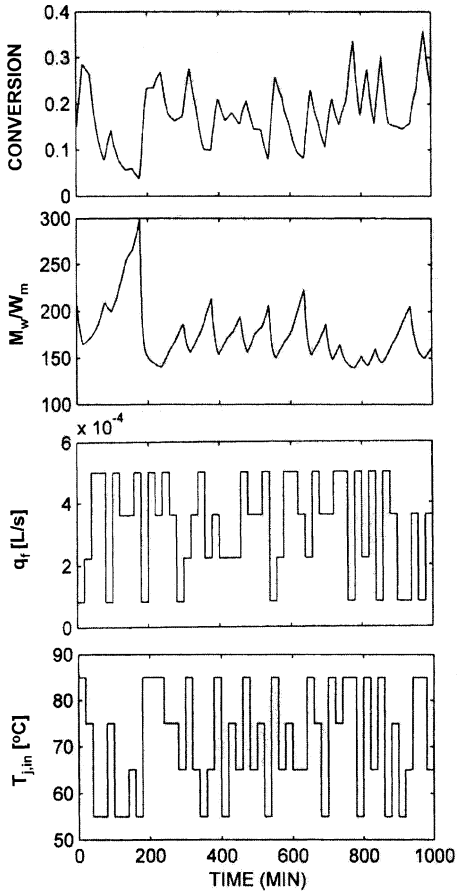

Figure 2. Input and output data sequences for identifying controller parameters L_1 and L_2.

the high heat-transfer efficiency. Hence, the jacket inlet temperature is good enough to capture the nonlinearity of the reactor system.

Simulation results

The input/output data-based predictive control was applied to the styrene solution polymerization reactor. Figure 2 presents the input/output data sequence used for the identification of the controller parameters L_1 and L_2. The sampling period is 1 min. The input signals were drawn from a uniform distribution with four levels. The pseudorandom four-level input signals excite nonlinear modes that pseudorandom binary signals (PRBS) cannot (Na and Rhee, 2000). Because of these features, we use the four-level input signals rather than the PRBS. The output data against the pseudorandom four-level input signals are obtained by the mathematical model. In Figure 2, the lower two diagrams show the inputs used in the identification of L_1 and L_2, while the upper two diagrams present the outputs calculated by the nonlinear reactor model.

In the upper two diagrams of Figure 3, the profiles of the controlled outputs in the case of the linear input/output data-based predictive controller (Song et al., 2001) are presented when there are step changes in the setpoints for the conversion and the weight average molecular weight. Also, the corresponding profiles of the reactor temperature, the jacket inlet temperature, and feed flow rate are shown. The sampling time of the discrete time control is 1 min, and the prediction horizon and control horizon are chosen to be 15 and 7 min, respectively. The best controller tuning parameters have been selected by trial-and-error methods. Here we

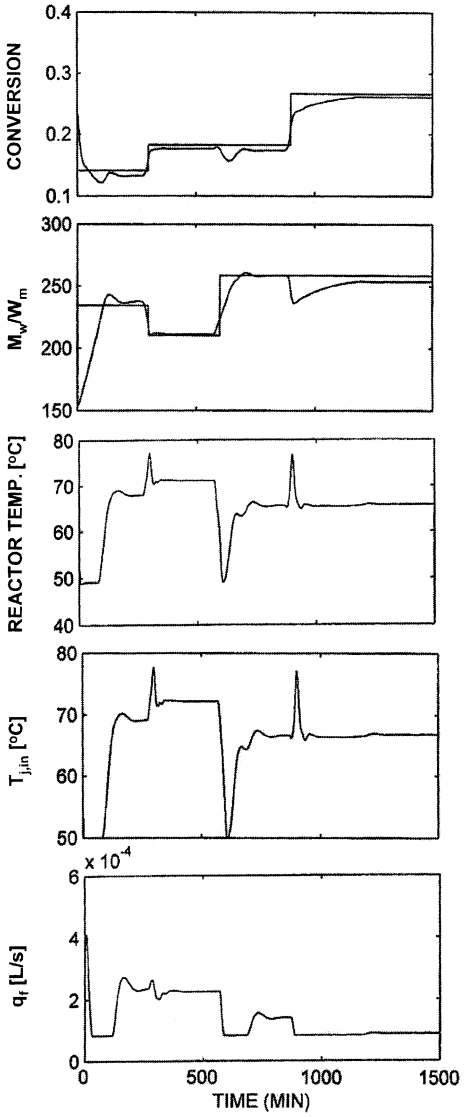

Figure 3. Servo performance of the linear input/output data-based predictive controller for step changes in the setpoints for conversion and molecular weight: Q = diag(1,1) and R_Δ = diag(1,1).

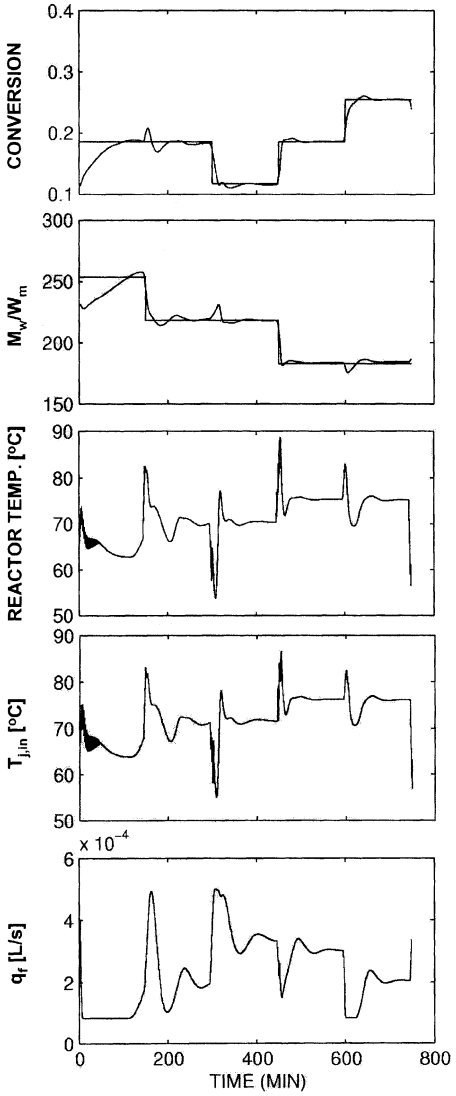

Figure 4. Servo performance of the LF-LPV input/output data-based predictive controller for step changes in the setpoints for conversion and molecular weight: Q = diag(5,5) and R_Δ = diag(2,2).

observe the offsets in both the conversion and the weight average molecular weight for a certain setpoints pair. This has happened because the linear input/output data-based predictive controller cannot suitably deal with the nonlinear features of the polymerization reactor system (Song et al., 2001).

Figure 4 shows the performance of the LF-LPV input/output data-based predictive controller for step changes in the setpoints for the conversion and the weight average molecular weight. The prediction horizon and control horizon are chosen to be 6 and 3 min, respectively. From this figure, we observe that the LF-LPV input/output data-based predictive controller performs satisfactorily without offset for the control of the conversion and the weight average molecular weight. This indicates that the LF-LPV model structure can deal with the nonlinear characteristics of the polymerization reactor efficiently.

Conclusions

In this work, as an alternative to "full-fledged" nonlinear model-predictive control, we introduce a different model-predictive control algorithm that uses an LF-LPV prediction model, and a subspace identification algorithm for LF-LPV models is reformulated from the control point of view. The proposed LF-LPV model-based controller is a sort of gain-scheduled control algorithm for nonlinear plants, and we use such heuristic design rules as "the scheduling parameter must capture the nonlinearity of the system and be on-line measurable to construct the control system." Those are common requirements for the gain-scheduled controller.

We refer to the proposed algorithm as the input/output data-based predictive controller, in which an explicit model of the system to be controlled is not calculated at any point in the algorithm. The proposed algorithm allows for the construction of a nonlinear model-predictive controller of an unknown nonlinear system, directly from a set of open-loop measurements of that system. As an illustrative example of the input/output data-based predictive control, we consider the styrene solution polymerization in a continuous reactor system and prove the superior performance of the LF-LPV input/output data-based predictive controller for polymer quality control.

The proposed approach gives a new angle for attacking the problem of identifying and controlling nonlinear systems. It is evident that the design method needs to be analyzed more thoroughly. For instance, the problem of controller reduction and the closed-loop scheme are needed from a practical point of view. The stability issue is also very important.

Acknowledgment

The financial aid from the Brain Korea 21 Program supported by the Ministry of Education is gratefully acknowledged.

Literature Cited

Bose, N. K., *Applied Multidimensional Systems Theory*, Van Nostrand, New York (1982).

Hamer, J. W., T. A. Akarmov, and W. H. Ray, "The Dynamic Behavior of Continuous Polymerization: II. Nonisothermal Solution Homopolymerization and Copolymerization in a CSTR," *Chem. Eng. Sci.*, **36**, 1897 (1981).

Hjalmarsson, H., M. Gevers, S. Gunnarsson, and O. Lequin, "Iterative Feedback Tuning: Theory and Applications," *IEEE Control Sys. Mag.*, **18**, 26 (1998).

Lee, J. H., "Modeling for Nonlinear Model Predictive Control: Requirements, Current Status and Future Research Needs," *Nonlinear Model Predictive Control*, F. Allgower and A. Zheng, eds., Birkhauser Verlag, Basel (2000).

Leith, D. J., and W. E. Leithead, "Gain-Scheduled Nonlinear Systems: Dynamic Analysis by Velocity-Based Linearisation Families," *Int. J. Control*, **70**, 289 (1998).

Na, S. S., and H.-K. Rhee, "Polynomial ARMA Model Identification for a Continuous Styrene Polymerization Reactor Using On-Line Measurements of Polymer Properties," *J. Appl. Poly. Sci.*, **76**, 1889 (2000).

Na, S. S., and H.-K. Rhee, "An Experimental Study for Property Control in a Continuous Styrene Polymerization Reactor Using a Polynomial ARMA Model," *Chem. Eng. Sci.*, **57**, 1165 (2002).

Shamma, J. S., and M. Athans, "Guaranteed Properties of Gain Scheduled Control for Linear Parameter-Varying Plants," *Automatica*, **27**, 559 (1991).

Shamma, J. S., and M. Athans, "Gain Scheduling: Potential Hazards and Possible Remedies," *IEEE Control Syst. Mag.*, **13**, 101 (1992).

Song, I. H., K. Y. Yoo, and H.-K. Rhee, "Analysis and Design of a Linear Input/Output Data-Based Predictive Control," *Ind. Eng. Chem. Res.*, **40**, 4292 (2001).

Van Overschee, P., and B. de Moor, *Subspace Identification for Linear Systems, Theory-Implementation-Applications*, Kluwer, Boston (1996).

Yoo, K. Y., *Subspace Identification and Control of Linear-Fractional LPV Systems*, PhD Thesis, Seoul National Univ., Seoul, Korea (2000).

Yoo, K. Y., B. G. Jeong, and H.-K. Rhee, "Molecular Weight Distribution Control in a Batch Polymerization Reactor Using the On-Line Two-Step Method," *Ind. Eng. Chem. Res.*, **38**, 4805 (1999).

Appendix: Orthogonal and Oblique Projections

Orthogonal and oblique projections of the row space of matrices are a commonly used tool in subspace identification. Suppose we are interested in the projection of the row space of $\mathfrak{A} \in \mathbb{R}^{p \times j}$, $\mathfrak{B} \in \mathbb{R}^{q \times j}$, and $\mathfrak{C} \in \mathbb{R}^{r \times j}$.

• *Orthogonal projection:* The orthogonal projection of the row space of $\mathfrak{A}$ into the row space of $\mathfrak{B}$ is defined as

$$\mathfrak{A}/\mathfrak{B} \equiv \mathfrak{A}\mathfrak{B}^{\dagger}\mathfrak{B} = L_{\mathfrak{B}}\mathfrak{B}$$

• *Oblique projection:* The oblique projection of the row space of $\mathfrak{A}$ into the row space of $\mathfrak{B}$ along the row space of $\mathfrak{C}$ is denoted by $\mathfrak{A}/_{\mathfrak{C}}\mathfrak{B}$. This oblique projection can be interpreted through the following recipe: Project the row space of $\mathfrak{A}$ orthogonally on the joint row space of $\mathfrak{B}$ and $\mathfrak{C}$ and decompose the result along the row space of $\mathfrak{B}$. This leads to the following definition for the oblique projection

$$\mathfrak{A}/_{\mathfrak{C}}\mathfrak{B} \equiv \mathfrak{A}\left[\begin{pmatrix}\mathfrak{B}\\ \mathfrak{C}\end{pmatrix}^{\dagger}\right]_{(:,1:q)}\mathfrak{B}$$

We also have

$$\mathfrak{A}/\begin{pmatrix}\mathfrak{B}\\ \mathfrak{C}\end{pmatrix} = \mathfrak{A}/_{\mathfrak{C}}\mathfrak{B} + \mathfrak{A}/_{\mathfrak{B}}\mathfrak{C}$$

Manuscript received Sept. 13, 2000, and revision received Feb. 4, 2002.