MODEL IDENTIFICATION AND ERROR COVARIANCE MATRIX ESTIMATION FROM NOISY DATA USING PCA

Shankar Narasimhan *,[1] and **Sirish L. Shah** *

* *Department of Chemical & Materials Engineering, University of Alberta, Edmonton, AB, CANADA*

Abstract: Principal Components Analysis (PCA) is increasingly being used for reducing the dimensionality of multivariate data, process monitoring, model identification, and fault diagnosis. However, in the mode that PCA is currently used, it can be statistically justified only if measurement errors in different variables are assumed to be i.i.d. In this paper, we develop the theoretical basis and an iterative algorithm for model identification using PCA, when measurement errors in different variables are unequal and are correlated. The proposed approach not only gives accurate estimates of both the model and error covariance matrix, but also provides answers to the two important issues of data scaling and model order determination.

Keywords: PCA, model identification, measurement errors, data scaling

1. INTRODUCTION

Principal Components Analysis is a multivariate statistical tool developed primarily to obtain a parsimonious representation of multivariate data. This is achieved by choosing a few linear combinations known as principal components, which together capture most of the variability in the data. The number of linear combinations chosen is typically less than the number of measured variables. In chemical engineering, PCA has been used in a similar manner for data compression. In recent years, PCA is also gaining significant importance as a tool for model identification or to discover the underlying spatial and/or temporal relationships between variables. For example, PCA is a critical part of many subspace based dynamic model identification methods (Viberg, 1995). The model identified using PCA has also been subsequently used in fault diagnosis (Yoon and MacGregor, 2000).

If measurements are corrupted by random errors, then PCA is an optimal procedure for estimating the model parameters only if the errors in different variables are independently and identically distributed (Wentzell *et al.*, 1997). An improved approach, called the maximum likelihood PCA (MLPCA), has been developed by Wentzell *et al.* (1997) for general error covariance matrix structures. However, their method assumes that the measurement error covariance matrix is known. It would be advantageous if the measurement error covariance matrix can be estimated along with the model from the same data set. This becomes especially important in chemical processes, since the model as well as the error covariance matrix are likely to change over time.

In this paper, we describe an iterative method which combines PCA with a maximum likelihood estimation procedure for obtaining an estimate

[1] Visiting Professor from Department of Chemical Engineering, IIT Madras, Chennai, INDIA

of the error covariance matrix. The proposed approach also provides answers to important questions on how to scale measured data before applying PCA, and how to obtain the model order without a priori knowledge.

2. MODEL IDENTIFICATION USING PCA WITH NOISE FREE DATA

We first discuss the case of model identification using PCA when the measurements are not corrupted by noise. Although this is well known, we present an alternative viewpoint which motivates the development of our proposed approach. We will consider the following process identification problem, which, despite its simplicity, contains the essential features for describing more complex processes.

Let $x(t)$ be a set of n variables at time instant t, which are related by the following set of m independent linear constraints

$$Ax(t) = 0 \tag{1}$$

where $A : m \times n$ is a constant time invariant constraint matrix. The above equations represent the spatial relations between variables, which are assumed to hold at all time instants. At each time instant, measurements $y(t)$ of all the variables corrupted by random errors are available, which can be written as

$$y(t) = x(t) + \epsilon(t) \tag{2}$$

We assume that the random errors, $\epsilon(t)$, are temporally independent and follow a multivariate normal distribution with mean zero and covariance matrix Σ_ϵ. The random errors are also assumed to be independent of $x(t)$. Given a sample of N measurements, $y(1) \dots y(N)$, the objective is to estimate the constraint matrix (also referred to as the model).

We assume that the true values of variables, $x(t)$, are a deterministic sequence satisfying the following two conditions.

$$\lim_{N\to\infty} \{\sqrt{N}(\bar{x} - \mu_x)\} = 0$$

$$\lim_{N\to\infty} \sqrt{N} \left[\sum_{i=1}^{N} (x(i) - \mu_x)(x(i) - \mu_x)^T - \Sigma_x \right] = 0$$

where $\bar{x}$ represents the average of the sequence $x(t)$, and μ_x and Σ_x are bounded. The above assumptions ensure that $y(t)$ is a quasi-stationary signal (Ljung, 1999).

It can be easily observed that due to the constraints, the vectors $x(t)$ span a $n-m$ dimensional subspace of R^n (denoted as V_x). Furthermore, the rows of A span a m dimensional subspace of R^n (denoted as V_c), which is orthogonal to V_x. Thus, given a sample of measurements in R^n, the objective of model identification can be viewed as the problem of decomposing R^n into two orthogonal subspaces, one of which defines V_x and the other V_c. It can be further noted that in order to define V_x and V_c, we only need to identify a basis for each of these spaces. Thus for identifying the model, it is sufficient to estimate any m linearly independent vectors in the row space of A.

In the absence of measurement errors, if we have a sample of $n-m$ linearly independent realizations of $x(t)$, then we can use it as a basis for V_x. We can then construct m linearly independent vectors orthogonal to V_x, which define a basis for V_c exactly. Note that this is sufficient to solve the stated problem.

If we use PCA to solve the above problem, then we will determine the orthonormal eigenvectors of the data variance matrix $S_y = \frac{1}{N} Y^T Y$ (which is identical to $S_x = \frac{1}{N} X^T X$ in the absence of measurement errors), where

$$Y = [y(1), y(2), \dots, y(N)]^T \tag{3}$$

Since the column space of S_y is identical to V_x, the matrix S_y has rank $n-m$. Thus, it will have $n-m$ nonzero eigenvalues, while the rest are zeros. The eigenvectors corresponding to the nonzero eigenvalues is an orthonormal basis for V_x. These eigenvectors are linear combinations of the variables x_i, and are called principal component directions. The eigenvector corresponding to the largest eigenvalue is the direction in V_x of maximum variability, and so on, in decreasing order of the magnitudes of the eigenvalues. The transpose of the m eigenvectors corresponding to the zero eigenvalues represent a basis for V_c. Note that these eigenvectors are not uniquely defined, because the corresponding eigenvalues are all equal. Although in some applications the PC directions may be useful, from the viewpoint of model identification they do not have any advantage over any other basis choice. Nevertheless, PCA does identify a basis for V_c exactly in the absence of measurement errors.

3. EFFECT OF SCALING IN PCA

We can raise the question of whether we can obtain an exact basis for V_c in the absence of measurement errors, if we scale the data before applying PCA. In order to answer this question, we will consider the following general linear transformation of the data

$$y_s(t) = Dy(t) = Dx(t) = x_s(t) \tag{4}$$

where D is any nonsingular matrix. If D is diagonal, then the above transformation defines a

scaling of the data. We can apply PCA to the variance matrix $S_{y_s} = \frac{1}{N} Y_s^T Y_s$ where the scaled data matrix Y_s is defined in a manner analogous to eq. 3. Since D is nonsingular, the rank of S_{y_s} is also equal to $n-m$. Thus, if we apply PCA using S_{y_s}, the transpose of the m orthonormal eigenvectors corresponding to the zero eigenvalues represent a basis for the space orthogonal to the scaled data vector $x_s(t)$. If we denote the transpose of these eigenvectors by A_s, then we can write

$$A_s x_s(t) = 0 \tag{5}$$

Using eq. 4 in the above equation we get

$$A_s D x(t) = 0 \tag{6}$$

From the above equation, we can deduce that the rows of the matrix $A = A_s D$ is a basis for V_c. Thus, in the absence of measurement errors, we obtain an exact basis for V_c even if we apply PCA to transformed (or scaled) data using eq. 4. However, it must be noted that the rows of A are not orthonormal and they also do not correspond to the eigenvectors of S_y.

4. MODEL IDENTIFICATION WITH KNOWN Σ_ϵ

We now consider the problem of model identification from noisy measurements using PCA, under the assumption that the measurement error covariance matrix, Σ_ϵ, is known. If measurements are noisy, then S_y will be a full rank matrix, and by using PCA we will not be able to obtain an exact basis for V_x or V_c. In fact, it is not possible to establish a relationship between the orthonormal eigenvectors of S_y and those of S_x. Furthermore, if we scale or transform the data using eq. 4, the eigenvectors of S_{y_s} and those of S_y do not bear any simple relation to each other (Morrison, 1967). Both these problems have been hitherto tackled in a heuristic manner in model identification from noisy data using PCA. If we assume that the error variances are much smaller compared to the variances in $x(t)$, then we can expect S_y to possess $n-m$ dominant eigenvalues and m small eigenvalues. The orthonormal eigenvectors corresponding to the small eigenvalues can be used as an estimate for the basis of V_c. It has also been suggested that if x contains variables which are not commensurate, then it is better to scale the data using standard deviations of the measurements. Other scaling strategies have also been suggested which can be applied under restrictive assumptions (Wentzell *et al.*, 1997). The effect of these heuristics on the quality of the identified model cannot be easily assessed. In what follows, we describe a procedure which effectively resolves the issue of appropriately scaling noisy data, such that a basis for V_c can be exactly obtained using PCA, under the assumption that Σ_ϵ is known.

Let L be the square root of Σ_ϵ defined by

$$LL^T = \Sigma_\epsilon \tag{7}$$

Similar to eq. 4, we will transform the measurements using L^{-1} as the nonsingular transformation matrix. The transformed measurements are given by

$$\begin{aligned} y_s(t) = L^{-1} y(t) &= L^{-1} x(t) + L^{-1} \epsilon(t) \\ &= x_s(t) + L^{-1} \epsilon(t) \end{aligned} \tag{8}$$

If Σ_ϵ is a diagonal matrix, then L is also a diagonal matrix containing the standard deviations of measurement errors, and the above transformation is equivalent to scaling the data using standard deviations of the corresponding measurement errors.

By taking the expectation of S_{y_s}, it can be easily shown that

$$\Sigma_{y_s} = S_{x_s} + I \tag{9}$$

In the above equation Σ_{y_s} is the population variance matrix of y_s, while $S_{x_s} = L^{-1} S_x L^{-T}$ (since $x(t)$ is deterministic).

From eq. 9 and the Eigenvalue Shift Theorem, the following two important results can be immediately derived.

(1) The eigenvectors of Σ_{y_s} are identical to those of S_{x_s}.
(2) The eigenvalues of Σ_{y_s} are equal to the corresponding eigenvalues of S_{x_s} increased by unity.

Since S_{x_s} is of rank $n-m$ it will have m zero eigenvalues. From the above results, we can conclude that the corresponding eigenvalues of Σ_{y_s} will be unity. Furthermore, the eigenvectors, corresponding to the eigenvalues of Σ_{y_s} that are greater than unity, define a basis for V_{x_s}. We have already shown that we can obtain the basis for V_x exactly, given the basis for V_{x_s}. Thus, given a sample of measurements, we can transform the measurements as in eq. 8 and apply PCA on S_{y_s}. The eigenvectors corresponding to the eigenvalues that are close to unity, can be used to obtain a basis for V_c (refer to the discussion that follows eq. 6). Using Theorem 2.3 (Ljung, 1999) for a quasi-stationary signal, we can prove that S_{y_s} is a consistent estimate of Σ_{y_s}. Thus, in the limit as the sample size goes to infinity, an exact basis for V_c is obtained using this method.

Wentzell *et al.* (1997) proposed a maximum likelihood estimation technique for model identification using PCA when the covariance matrix of measurement errors is known, and the model order is also specified. Their procedure is an alternating regression procedure which does not scale the data. Instead, it iteratively transforms

the model identified by PCA on unscaled data, until the maximum likelihood estimates of $x(t)$ are obtained. In contrast, the procedure we have described above is a non-iterative technique, which has a stronger theoretical basis and also provides additional useful information. In particular, the fact that the eigenvalues of S_{y_s} corresponding to the eigenvectors which define a basis for V_c should be unity, can be used to obtain the model order m. If an incorrect value of m is assumed, then the eigenvalues corresponding to the last m eigenvectors of S_{y_s} may not be close to unity.

5. SIMULTANEOUS MODEL IDENTIFICATION AND ERROR COVARIANCE MATRIX ESTIMATION

If Σ_ϵ is unknown, then the method described in the preceding section can be applied, if we can estimate the error covariance matrix from the data along with the model. We describe an iterative algorithm for achieving this by combining PCA with a maximum likelihood estimation (MLE) method for obtaining an estimate of the error covariance matrix. We will assume that an initial estimate of the model constraint matrix, $\hat{A}^0$, is available. (Such an estimate can be obtained by applying PCA to the measured data). Using this initial model estimate, we compute the constraint residuals at each time instant as

$$r(t) = A^0 y(t) \tag{10}$$

If the estimated model is exact, then the constraint residuals will be independent normally distributed variables with zero mean and covariance matrix $\Sigma_r = \hat{A}^0 \Sigma_\epsilon (\hat{A}^0)^T$. Thus, the joint density function of $r(1) \dots r(N)$ can be easily obtained, and an estimate of Σ_ϵ can be obtained by maximizing the log likelihood function of $r(1) \dots r(N)$. This results in the following nonlinear optimization problem.

$$\min_{\Sigma_\epsilon} N \log |\hat{A}^0 \Sigma_\epsilon (\hat{A}^0)^T| + \sum_{i=1}^{N} (r_i^T(t) (\hat{A}^0 \Sigma_\epsilon (\hat{A}^0)^T)^{-1} r_i(t)) \tag{11}$$

The above MLE problem can also be interpreted as a procedure for extracting an estimate of Σ_ϵ, given an estimate of the covariance matrix of constraint residuals Σ_r. This follows from the fact that the maximum likelihood estimate of Σ_r (which maximizes the likelihood function of $r(1) \dots r(N)$) is the sample covariance matrix S_r. The estimate of Σ_ϵ, which maximizes the same likelihood function, is the one that satisfies the following relation.

$$\hat{A}^0 \hat{\Sigma}_\epsilon (\hat{A}^0)^T = S_r \tag{12}$$

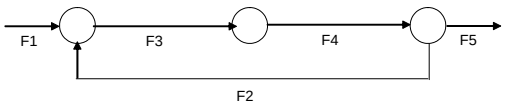


Fig. 1. Schematic of a flow process

Depending on the number of constraints and number of variables, it may or may not be possible to satisfy the above equation. Typically, the number of spatial relations m is usually less than n. In such cases, if we attempt to estimate all diagonal and off-diagonal elements of Σ_ϵ, multiple solutions that satisfy the above equation are obtained. One possibility is to assume that Σ_ϵ is diagonal, and estimate only the n diagonal elements corresponding to the measurement error variances. Even in this case, a non-degenerate estimate for Σ_ϵ is obtained only if $m(m+1) \geq 2n$. Other techniques have been proposed for estimating the measurement error covariance matrix, given the constraint model and the covariance matrix of constraint residuals (Romagnoli and Sanchez, 1999). However, these methods are not maximum likelihood estimates.

Assuming that the number of diagonal and off-diagonal elements of Σ_ϵ that we are estimating is less than or equal to $m(m+1)/2$, we can minimize (11). We can also impose lower bounds on the elements of Σ_ϵ that we are estimating, and solve the constrained optimization problem. Let us denote the estimate of measurement error covariance matrix obtained using the above method as $\hat{\Sigma}_\epsilon^0$. Note that this estimate has been obtained assuming that the model has been estimated exactly. We can use $\hat{\Sigma}_\epsilon^0$ to transform the data as described in the preceding section, and apply PCA on the transformed measurements to get an updated estimate of the constraint matrix. We can repeat the entire procedure until the estimates for the model and error covariance matrix converge. A simple test of convergence is to check that the singular values obtained using PCA do not change significantly from one iteration to the next.

6. SIMULATION RESULTS AND DISCUSSION

A flow process example shown in Fig. 1, has been chosen to test the proposed procedure. The above example has been chosen so that it satisfies the condition $m(m+1) > 2n$ (in the above example $m = 3$ and $n = 5$). We will assume that the measurement error covariance matrix is diagonal.

In order to simulate the true values of variables at each time instant, a set of independent flow variables are chosen (in the above example F1 and F2 are chosen as independent variables). The true values of independent variables are simulated by

adding normally distributed random fluctuations to their base values. The true values of the dependent flow variables are calculated such that they satisfy the flow balance constraints. The base values of variables and the standard deviations of the fluctuations are given in Table 1. In the simulation

Table 1. Data for simulating true values of variables.

Flow variable	True values		σ_ϵ
	Base value	Std of fluctuation	
F1	10	1.0	0.1
F2	10	2.0	0.08
F3	F1 + F2		0.15
F4	F3		0.2
F5	F4 - F2		0.18

procedure, the measured values of variables are simulated by adding normally distributed random noise to their true values. The standard deviations of measurement errors are also given in Table 1. A sample of 1000 measurement vectors is simulated and the procedure described in Section 5 is applied.

In order to evaluate the accuracy of the estimated basis for V_c, the distance between the row spaces of the true constraint matrix and the estimated constraint matrix can be used. The minimum distance of each row of A from the subspace spanned by the rows of $\hat{A}$ is given by

$$\alpha_i = ||A_{.i}^T - A_{.i}^T \hat{A}^T (\hat{A}\hat{A}^T)^{-1} \hat{A}|| \qquad (13)$$

A consolidated measure of model estimation accuracy is given by

$$\alpha = \sum_i \alpha_i \qquad (14)$$

The above measure treats all bases sets for the row space of $\hat{A}$ as equivalent. Alternatively, the angle θ between the row spaces of A and $\hat{A}$ can also be used as a measure of the model estimation accuracy.

The results obtained for the above example using PCA for different choices of data scaling and the proposed iterative method (denoted as IPCA) are presented in Table 2. In both approaches, the actual number of constraints are assumed to be known.

In Table 2, the first three rows are the results obtained using PCA, respectively, when the measured data are not scaled, scaled using sample standard deviation of the corresponding measurement, and scaled using true standard deviations of measurement errors. The last row gives the results obtained using the proposed method. The constraint matrix obtained by PCA is used as an initial estimate in IPCA. From the values of α and θ, we can conclude that a good estimate of the model constraints is obtained using both PCA

Table 2. Quality of the model identified for different scaling choices.

Case	Scale	$\alpha \times 10^3$	θ (deg)
PCA	None	5.86	0.17
PCA	σ_y	10.22	0.24
PCA	σ_ϵ	1.62	0.028
IPCA		1.2	0.03

and IPCA. This is due to the fact that in this simulation, the signal to noise variation is high (ratio of their standard deviations is more than 10). However, even in this case, the proposed iterative method is able to improve the accuracy of the model obtained through PCA by more than 80%. The number of major iterations required for IPCA to converge was around ten, although within three to four iterations the estimates obtained are very close to the final converged values. The estimated standard deviations of measurement error variances obtained using the proposed method are [0.1121 0.0837 0.1406 0.2031 0.1775], which are close to their true values. For the given sample of data, the best achievable model accuracy is obtained when the data are scaled using the true standard deviations of measurement errors, as shown in the third row of Table 2. It is observed that the accuracy of model obtained using IPCA is very close to this achievable limit.

The converged singular values, [236.5 17.7 1.01 1.0 0.99], obtained using IPCA reveal an interesting feature. It can be observed, that the singular values corresponding to the last three PCs (which correspond to the assumed number of constraints) are very close to unity, as theoretically predicted. In contrast, the singular values obtained using PCA for the three scaling strategies are, respectively, [33.32 1.9 0.18 0.16 0.11], [19.84 1.35 0.15 0.08 0.06], and [238.9 18.8 1.06 0.99 0.97]. Clearly by scaling the data differently, we can alter the singular values, and it may make it difficult to determine the number of PCs to be retained/rejected. In other words, we may not be able to determine the number of constraints precisely by examining the singular values of the scaled data, unless we use the standard deviations of measurement errors for scaling. It may also be noted that if the data is auto-scaled, a worse model may be obtained compared to the case when the data is not scaled at all (compare results of first and second rows of Table 2).

In order to evaluate how the proposed method performs for low signal to noise variation, the standard deviations of the true value variations in F1 and F2 are reduced to 0.2 each, while retaining the standard deviations of measurement errors as before. The results obtained for this case are given in Table 3. As expected the accuracy of the models estimated by both approaches has decreased. However, a good estimate of the model is still

Table 3. Quality of the model identified for low signal to noise ratio.

Case	Scale	$\alpha \times 10^3$	θ (deg)
PCA	None	447.0	12.73
PCA	σ_y	252.1	7.61
PCA	σ_ϵ	21.1	0.49
IPCA		32.5	1.39

obtained using the proposed approach, and there is a 90% improvement over the model obtained using PCA. The estimated standard deviations of measurement errors using the proposed approach are [0.1121 0.0838 0.1406 0.2031 0.1774], which are same as before. Thus even though the model is estimated less accurately, the measurement error standard deviations are estimated fairly accurately by the proposed approach. The converged singular values obtained are [233.5 2.4 1.01 1.0 0.98], which again satisfy the condition that the singular values corresponding to the assumed number of constraints are close to unity.

In order to demonstrate that our proposed method can be used even if errors in different variables are correlated, we simulated data for the above example using an error covariance structure which contained an off-diagonal element. It should be noted, that since the above process has only 3 constraints, we can estimate at most 6 elements of the error covariance matrix. This implies that besides the diagonal elements, at most one off-diagonal element can be estimated from the measured data. The true flow rates in this case are simulated as described in Table 1. The non-zero elements of the measurement error covariance are chosen as [0.0244 0.0064 0.0369 0.04 0.0324 0.03], where the first five elements are the diagonal elements (error variances) and the last element is the covariance between errors in variables 1 and 3. The results for this case are shown in Table 4.

Table 4. Model identification for non-diagonal error covariance matrix.

Case	Scale	$\alpha \times 10^3$	θ (deg)
PCA	None	10.53	0.20
PCA	σ_y	13.68	0.27
PCA	cholesky factor of Σ_ϵ	1.21	0.024
IPCA		1.47	0.043

The above results again indicate that the model obtained using IPCA is better than that obtained using PCA, and is close to the maximum achievable accuracy. The non-zero elements of the estimated error covariance matrix are [0.0266 0.007 0.0367 0.0402 0.0312 0.0312], which are also close to their corresponding true values. The converged singular values obtained using IPCA are [2458.5 27.53 1 1 1]. As theoretically predicted, the last three singular values are unity even in this case.

We had stated that it may also be possible to determine the number of constraints using the proposed approach. As a test of this, the number of constraints was incorrectly assumed as four instead of three in the above simulations, and the proposed method was used. In this case, the estimated std of measurement errors obtained are [1.118 1.117 0.047 0.242 1.127], and the singular values are [434.37 1.72 1.00 0.15 0.11]. Since the singular values corresponding to the last four eigenvalues are not close to unity, this indicates that the number of constraints has been incorrectly assumed.

7. CONCLUDING REMARKS

In this paper, we have proposed an algorithm for simultaneously estimating an accurate process model and the measurement error covariance matrix from noisy data, using an iterative PCA technique. As part of the development, the outstanding issue of appropriately scaling or transforming noisy data before applying PCA, has also been resolved. A new criteria for determining model order by examining the eigenvalues obtained using PCA on the transformed data is proposed, which has a rigorous theoretical basis. As part of future research, the technique described here can be extended to methods which use PCA as an integral component such as PCR, PLS, and subspace based model identification.

Acknowledgements: Financial support from NSERC, Matrikon Inc. and ASRA in the form of the Industrial Research Chair Program at the University of Alberta is gratefully acknowledged.

REFERENCES

Ljung, L. (1999). *System Identification: Theory for the User*. 2nd ed.. Prentice-Hall. New Jersey.

Morrison, D.F. (1967). *Multivariate Statistical Methods*. 2nd ed.. McGraw-Hill.

Romagnoli, J.A. and M.C. Sanchez (1999). *Data Processing and Reconciliation For Chemical Process Operation*. Academic Press.

Viberg, M. (1995). Subspace-based method for the identification of linear time-invariant systems. *Automatica* **31**(12), 1835–1851.

Wentzell, P.D., D.T. Andrews, D.C. Hamilton, K. Faber and B.R. Kowalski (1997). Maximum likelihood principal component analysis. *J. Chemometrics* **11**, 339–366.

Yoon, S. and J.F. MacGregor (2000). Statistical and causal model-based approaches to fault detection and isolation. *AIChE J.* **46**(9), 1813–1899.

NONLINEAR SUBSPACE MODEL IDENTIFICATION

Ali Cinar and Jeffrey DeCicco [1]

Illinois Institute of Technology
Department of Chemical and Environmental Engineering
10W 33rd Street, Chicago, IL, 60616 USA

Abstract: Canonical variates state space (CVSS) modeling is a popular subspace linear model identification technique. A nonlinear extension of CVSS modeling approach was proposed (DeCicco and Cinar, 2000). The modeling procedure consists of two steps: development of a multivariable nonlinear model for a set of latent variables and the linking of the latent variables to outputs of the process. The nonlinear model is structured like a Generalized Additive Model (GAM) and is estimated with CANALS, a nonlinear canonical variate analysis algorithm. This communication presents the methodology and an illustrative example of chemical reactor modeling using data generated from a detailed polymerization reactor model.

Keywords: subspace identification, nonlinear autoregressive model, chemical reactor model identification

1. INTRODUCTION

Canonical variate analysis (CVA) and canonical variate (CV) regression are powerful methods used for developing linear dynamic models. Most notably they are used in subspace modeling to estimate linear state space models (Larimore, 1990*b*). Subspace methods are attractive because of their ease in which they can model multivariate systems. An extension of linear CVA for finding nonlinear state space models was examined (Larimore, 1990*a*) where use of alternating conditional expectation (ACE) algorithm (Breiman and Friedman, 1985) was suggested as the nonlinear CVA method. The examples used linear CVA to model a system by augmenting the linear system with polynomials of past outputs.

Subspace modeling can be cast as a reduced rank regression (RRR) of collections of future outputs on past inputs and outputs after removing the effects of future inputs. CVA performs this RRR. In the case of a linear system, an approximate Kalman filter sequence is recovered from this regression. The state space coefficient matrices are recovered from the state sequence. The nonlinear approach extends this regression to allow for possible nonlinear transformations of the past inputs and outputs, and future inputs and outputs before RRR is performed. The model structure consists of two sub-models. The first model is a multivariable dynamic model for a set of latent variables, the second relates these latent variables to outputs. The latent variables are linear combinations of nonlinear transformations of past inputs and outputs. These nonlinear transformations or functions are found using CANALS (van der Burg and de Leeuw, 1983). Using nonlinear CVA to fit dynamic models is not new. ACE algorithm was used to visually infer nonlinear functions for single output additive models (Chen and Tsay, 1993). This work differs in that the nonlinear functions estimated are directly utilized for prediction. Also, a collection of multiple future outputs is consid-

[1] Corresponding author (e-mail : *cinar@iit.edu).*

ered, which leads to the latent variables model structure. The latent variables are then linked to the outputs using linear projection type nonlinear model structures such as projection pursuit regression (PPR) (Friedman and Stuetzel, 1981) or a linear model through least squares regression.

2. NONLINEAR MODEL STRUCTURE

Let $\mathbf{y}_t \in \mathcal{R}^l$, $\mathbf{u}_t \in \mathcal{R}^m$ and $\mathbf{x}_t \in \mathcal{R}^n$ represent outputs, inputs and latent variables that are collections of individual variables y_t^i, u_t^i and x_t^i,

$$\mathbf{y}_t = \begin{bmatrix} y_t^1 \\ \vdots \\ y_t^l \end{bmatrix}, \ \mathbf{u}_t = \begin{bmatrix} u_t^1 \\ \vdots \\ u_t^m \end{bmatrix}, \ \mathbf{x}_t = \begin{bmatrix} x_t^1 \\ \vdots \\ x_t^n \end{bmatrix}. \quad (1)$$

The model structure for a single latent variable is

$$x_{t+\beta}^i = \sum_{j=1}^{\beta} \sum_{k=1}^{m} h_{i,j+k-1}^* \theta_{k,j}^p \left(u_{t+\beta-j}^k\right) + \sum_{j=1}^{\beta} \sum_{k=1}^{l} h_{i,\beta m+j+k-1}^* \phi_{k,j}^p \left(y_{t+\beta-j}^k\right) \quad (2)$$

where h^* are scalar coefficients, θ^p and ϕ^p are nonlinear functions, and β is the past window length. The model structure linking the latent variables to a single output is

$$y_t^i = y_{ss}^i + \sum_{j=1}^{M} n_{i,j} \psi_j \left(\mathbf{l}_j^T \mathbf{x}_t\right), \quad (3)$$

where $n_{i,j}$ are scalar coefficients, $\mathbf{l}_j$ are $n \times 1$ coefficient vectors, ψ_j is some nonlinear function, and y_{ss}^i is a steady state operating point.

The structure for the latent variables model in (2) is of a generalized additive model (GAM) (Hastie and Tibshirani, 1990). But it is developed using a nonlinear CVA method discussed in Section 4.

3. LINEAR MODELING

A number of methods fall under this framework including CVA (Larimore, 1990*b*) and N4SID (Van Overschee and De Moor, 1994). An estimate of Kalman filter states can be recovered from a RRR of a collection of future outputs on past collection of inputs and outputs, and future inputs in linear systems (Van Overschee and De Moor, 1994). "Past" and "future" discriminate previously observed historical data used in the estimation of a causal dynamic model.

Define the collection of past and future observed outputs:

$$\mathbf{y}_\beta(t) = [\mathbf{y}_t^T \ \mathbf{y}_{t+1}^T \ \cdots \ \mathbf{y}_{t+\beta-1}^T]^T, \quad (4)$$
$$\mathbf{y}_\gamma(t) = [\mathbf{y}_{t+\beta} \ \mathbf{y}_{t+\beta+1}^T \ \cdots \ \mathbf{y}_{t+\alpha-1}^T]^T,$$

where $\alpha = \beta + \gamma$ and γ is the future window length. The collections of past and future inputs ($\mathbf{u}_\beta$ and $\mathbf{u}_\gamma$) are defined in the same manner. Let $t+\beta-1$ represent the present time. Observations of the above collections are

$$\mathbf{Y}_\beta = [\mathbf{y}_\beta(1) \ \cdots \ \mathbf{y}_\beta(N)] \quad (5)$$
$$\mathbf{Y}_\gamma = [\mathbf{y}_\gamma(1) \ \cdots \ \mathbf{y}_\gamma(N)]$$

Collections of observed past and future inputs are $\mathbf{U}_\beta$ and $\mathbf{U}_\gamma$, respectively. The objective is to extract a causal model that predicts future outputs using assigned future input values:

$$\mathbf{Y}_\gamma = \mathbf{L_1}\mathbf{U}_\beta + \mathbf{L_2}\mathbf{Y}_\beta + \mathbf{L_3}\mathbf{U}_\gamma \quad (6)$$

where $\mathbf{L}_1$, $\mathbf{L}_2$, and $\mathbf{L}_3$ are coefficient matrices.

The RRR problem that leads to approximate state variables sequence is formulated as

$$\min_{\mathbf{L}_1, \mathbf{L}_2, \mathbf{\Phi}_\gamma} \| \mathbf{Y}_\gamma - [\mathbf{L_1}\mathbf{U}_\beta + \mathbf{L_2}\mathbf{Y}_\beta + \mathbf{L_3}\mathbf{U}_\gamma] \|_{\mathbf{W}}^2 \quad (7)$$

where $\| \mathbf{M} \|_{\mathbf{W}} = \mathrm{tr}\left(\mathbf{M}^T\mathbf{W}\mathbf{M}\right)$. The regression problem has a reduced rank structure in that $[\mathbf{L}_1 \ \mathbf{L}_2] = \mathbf{\Gamma}\mathbf{H}$, where $\mathbf{\Gamma}$ and $\mathbf{H}$ both have rank n. The CVA RRR solution is derived by setting $\mathbf{W} = \left(\mathbf{Y}_\gamma \mathbf{\Pi}_\gamma^\perp \mathbf{Y}_\gamma^T\right)^{-1}$ (Jansson and Wahlberg, 1999). Estimates of $\mathbf{\Gamma}$ and $\mathbf{H}$ are obtained from singular value decomposition (SVD)

$$\mathbf{W}^{\frac{1}{2}} \mathbf{Y}_\gamma \mathbf{\Pi}_\gamma^\perp \mathbf{P}_\beta^T \left(\mathbf{P}_\beta \mathbf{\Pi}_\gamma^\perp \mathbf{P}_\beta^T\right)^{-\frac{1}{2}} = \mathbf{Q}\mathbf{\Sigma}\mathbf{S}^T \quad (8)$$

$$\mathbf{\Pi}_\gamma^\perp = \mathbf{I} - \mathbf{U}_\gamma^T \left(\mathbf{U}_\gamma \mathbf{U}_\gamma^T\right) \mathbf{U}_\gamma, \quad \mathbf{P}_\beta = \begin{bmatrix} \mathbf{U}_\beta \\ \mathbf{Y}_\beta \end{bmatrix} \quad (9)$$

The real matrices $\mathbf{Q}$ and $\mathbf{S}$ contain the canonical variate vectors, and $\mathbf{\Sigma}$ contains the canonical correlations of the CVA decomposition (8). The estimates become

$$\mathbf{\Gamma} = \mathbf{W}^{-\frac{1}{2}} \times \left(\text{first } n \text{ columns of } \mathbf{Q}\mathbf{\Sigma}^{\frac{1}{2}}\right), \quad (10)$$
$$\mathbf{H} = \left(\text{first } n \text{ rows of } \mathbf{\Sigma}^{\frac{1}{2}}\mathbf{S}^T\right) \left(\mathbf{P}_\beta \mathbf{\Pi}_\gamma^\perp \mathbf{P}_\beta^T\right)^{-\frac{1}{2}}.$$

The approximate state variables sequence is

$$\mathbf{X} = \mathbf{H} \begin{bmatrix} \mathbf{U}_\beta \\ \mathbf{Y}_\beta \end{bmatrix}, \quad (11)$$

with $\mathbf{X} = [\hat{\mathbf{x}}(\beta+1) \ldots \hat{\mathbf{x}}(\beta+N)]$. With the estimated state variables sequence (11) and the observed inputs and outputs it is possible to estimate the model coefficients of the linear state space model including the Kalman filter gain (Larimore, 1990*b*; Van Overschee and De Moor, 1994).

4. NONLINEAR MODEL IDENTIFICATION

The nonlinear model identification is an extension of the linear approach. First nonlinear transformations of observed data are sought in the regression

of future outputs ($\mathbf{Y}_\gamma$) on past inputs and outputs, and future inputs ($\mathbf{U}_\beta$, $\mathbf{Y}_\beta$, and $\mathbf{U}_\gamma$). This leads to a set of latent variables that are nonlinear functions of past inputs and outputs. Then, the reduced rank structure and coefficient matrices are estimated using CVA as in the linear case. Finally, the model structure between latent variables and outputs is developed using projection pursuit (Friedman and Stuetzel, 1981) or other methods.

4.1 *Latent Variable Model Identification*

The nonlinear transformations are estimated by a modified version of the nonlinear CVA technique CANALS (van der Burg and de Leeuw, 1983). Once the nonlinear transformations are found, linear CVA is used to find the linear combinations that form the latent variable model. The regression model (6) is first generalized as

$$\mathbf{Z}_\gamma = \mathbf{L}_1\mathbf{V}_\beta + \mathbf{L}_2\mathbf{Z}_\beta + \mathbf{L}_3\mathbf{V}_\gamma, \tag{12}$$

where $\mathbf{L}_1$, $\mathbf{L}_2$, and $\mathbf{L}_3$ are coefficient matrices, $\mathbf{Z}_\gamma$, $\mathbf{Z}_\beta$, $\mathbf{V}_\gamma$, and $\mathbf{V}_\beta$ are nonlinear transformations of the past and future inputs and outputs:

$$\begin{aligned}
\mathbf{Z}_\gamma &= [\mathbf{z}_\gamma(1) \ \dots \ \mathbf{z}_\gamma(N)]^T \\
\mathbf{Z}_\beta &= [\mathbf{z}_\beta(1) \ \dots \ \mathbf{z}_\beta(N)]^T \\
\mathbf{V}_\gamma &= [\mathbf{v}_\gamma(1) \ \dots \ \mathbf{v}_\gamma(N)]^T \\
\mathbf{V}_\beta &= [\mathbf{v}_\beta(1) \ \dots \ \mathbf{v}_\beta(N)]^T \\
\mathbf{z}_\gamma(t) &= \left[\Phi_1^f(\mathbf{y}_{t+\beta})^T \ \dots \ \Phi_\gamma^f(\mathbf{y}_{t+\alpha-1})^T\right]^T \\
\mathbf{z}_\beta(t) &= \left[\Phi_\beta^p(\mathbf{y}_t)^T \ \dots \ \Phi_1^p(\mathbf{y}_{t+\beta-1})^T\right]^T \\
\mathbf{v}_\gamma(t) &= \left[\Theta_1^f(\mathbf{u}_{t+\beta})^T \ \dots \ \Theta_\gamma^f(\mathbf{u}_{t+\alpha-1})^T\right]^T \\
\mathbf{v}_\beta(t) &= \left[\Theta_\beta^p(\mathbf{u}_t)^T \ \dots \ \Theta_1^p(\mathbf{u}_{t+\beta-1})^T\right]^T
\end{aligned}$$

Φ and Θ are vector valued functions of the form

$$\Phi_i^p(y_t) = \begin{bmatrix} \phi_{1,i}^p(y_t^1) \\ \vdots \\ \phi_{l,i}^p(y_t^l) \end{bmatrix}, \quad \Theta_i^p(u_t) = \begin{bmatrix} \theta_{1,i}^p(u_t^1) \\ \vdots \\ \theta_{m,i}^p(u_t^m) \end{bmatrix}.$$

The observed inputs and outputs have been centered around their means or some steady state operating point of the process.

CANALS was originally developed for analysis of categorical data (van der Burg and de Leeuw, 1983) and did not utilize a locally adaptive regression technique. A modified version of CANALS estimates the nonlinear functions with nonparametric regression (DeCicco and Cinar, 2000). The final nonparametric estimates are interpolated by Chebychev polynomials to allow a smooth interpolation. The modified CANALS is used to estimate the coefficient matrices and nonlinear transformations. CANALS seeks to minimize

$$\min_{\tilde{\mathbf{L}},\mathbf{L}_4,\mathbf{G},\mathbf{Z}_\gamma} \| \mathbf{L}_4\mathbf{Z}_\gamma - \tilde{\mathbf{L}}\mathbf{G} \|^2 \tag{13}$$

where $\mathbf{L}_4$ is a canonical variate coefficient matrix, $\tilde{\mathbf{L}} = [\mathbf{L}_1 \ \mathbf{L}_2 \ \mathbf{L}_3]$, and $\mathbf{G}^T = \left[\mathbf{V}_\beta^T \ \mathbf{Z}_\beta^T \ \mathbf{V}_\gamma^T\right]$. The loss function (13) is minimized such that the nonlinear functions have zero mean and unit variance with the constraints:

$$\begin{aligned}
\mathbf{L}_4\tilde{\mathbf{Z}}_\gamma\left(\mathbf{L}_4\tilde{\mathbf{Z}}_\gamma\right)^T &= (N+\alpha-1)\mathbf{I}, \\
\tilde{\mathbf{L}}\mathbf{G}\left(\tilde{\mathbf{L}}\mathbf{G}\right)^T &= (N+\alpha-1)\mathbf{I},
\end{aligned}$$

where $\mathbf{I}$ is the identity matrix.

CANALS uses alternating least squares (ALS) to estimate coefficient matrices and nonlinear transformations. The ALS method works iteratively until convergence of (13). $\mathbf{L}_4$ and $\tilde{\mathbf{L}}$ are estimated by CVA RRR. The nonlinear transformations are estimated using a back-fitting approach (Hastie and Tibshirani, 1990). The supersmoother of (Friedman, 1984) is used to estimate the nonlinear functions in the back-fitting step.

The number of latent variables and linear combinations of nonlinear functions that make up these latent variables are determined by using linear CVA between the $\mathbf{Z}_\gamma$ and $\tilde{\mathbf{P}}_\beta = \left[\mathbf{V}_\beta^T \ \mathbf{Z}_\beta^T\right]^T$ after the effect of $\mathbf{V}_\gamma$ is removed. Use SVD

$$\left(\mathbf{Z}_\gamma\tilde{\mathbf{\Pi}}_\gamma^\perp\mathbf{Z}_\gamma^T\right)^{-\frac{1}{2}}\mathbf{Z}_\gamma\tilde{\mathbf{\Pi}}_\gamma^\perp\tilde{\mathbf{P}}_\beta^T\left(\tilde{\mathbf{P}}_\beta\tilde{\mathbf{\Pi}}^\perp\tilde{\mathbf{P}}_\beta^T\right)^{-\frac{1}{2}} = \mathbf{Q}\mathbf{\Sigma}\mathbf{S}^T,$$

$$\tilde{\mathbf{\Pi}}_\gamma^\perp = \mathbf{I} - \mathbf{V}_\gamma^T\left(\mathbf{V}_\gamma\mathbf{V}_\gamma^T\right)^{-1}\mathbf{V}_\gamma. \tag{14}$$

The linear combinations of estimated states are

$$\mathbf{H}^* = \left(\text{first } n \text{ rows of } \mathbf{\Sigma}^{\frac{1}{2}}\mathbf{S}^T\right) \times \left(\tilde{\mathbf{P}}_\beta\tilde{\mathbf{\Pi}}_\gamma^\perp\tilde{\mathbf{P}}_\beta^T\right)^{-\frac{1}{2}}.$$

The latent variable sequence becomes

$$\tilde{\mathbf{X}}(t) = \mathbf{H}^*\tilde{\mathbf{P}}_\beta, \tag{15}$$

where $\tilde{\mathbf{X}}(t) = [\tilde{\mathbf{x}}_{t+\beta} \ \tilde{\mathbf{x}}_{t+\beta+1} \dots \tilde{\mathbf{x}}_{t+\beta+N-1}]$. This leads to the latent variable model structure

$$\tilde{\mathbf{x}}(t+\beta) = \mathbf{H}^*\begin{bmatrix} \mathbf{v}_\beta(t) \\ \mathbf{z}_\beta(t) \end{bmatrix} \tag{16}$$

where $\mathbf{h}_i^*$ is the ith column of $\mathbf{H}^*$, and (16) is a generalization of (2). The number of latent variables n is chosen by inspecting the singular values of $\mathbf{\Sigma}$ in (14). Significant latent variables have relatively large singular singular values.

4.2 *Link Function*

The relationship between latent variables and outputs is generalized by a PPR model. This type

of model structure includes linear least squares and GAM model structures. We build a linear model estimated by CVA regression and a PPR model, and compare their performances. The linear model is a special case of the PPR structure. If a linear model is adequate the overall model structure is simpler. The multivariable version of (3) relating latent variables to outputs is

$$\mathbf{y}_t = \mathbf{y}_{ss} + \sum_{i=1}^{M} \mathbf{n}_i \psi_i \left(\mathbf{l}_i^T \mathbf{x}_t \right). \qquad (17)$$

PPR seeks to minimize the loss function

$$SSE = \sum_{j=1}^{l} \sum_{t=1}^{N+\alpha-1} \left(y_t^j - \sum_{i=1}^{M} n_{i,j} \psi_i (\mathbf{l}_i^T \mathbf{x}_t) \right)^2 \qquad (18)$$

with respect to $n_{i,j}$, $\mathbf{l}_i$, ψ_i, and M. A back-fitting procedure is utilized to perform the regression. First, $\mathbf{l}_i$ is found by minimizing (18) using numerical optimization. Next, ψ_i is found by nonparametric regression and lastly $\mathbf{n}_i$ is estimated with CVA regression. The number of terms M is chosen by using (18) (See Figure 1).

With the estimated latent variable sequence $\tilde{\mathbf{X}}(1)$ of (15) the loss function (18) may be written as

$$\min_{\mathbf{N},\mathbf{\Psi}\left(\mathbf{L}_x(\bar{\mathbf{X}}(1))\right)} \| \mathbf{Y}_\gamma[1,:] - \mathbf{N}\mathbf{\Psi}\left(\mathbf{L}_x^T \tilde{\mathbf{X}}(1)\right) \|^2 \qquad (19)$$

$$\mathbf{N} = [\mathbf{n}_1 \ \mathbf{n}_2 \ \ldots \ \mathbf{n}_M], \quad \mathbf{L}_x = [\mathbf{l}_1 \ \mathbf{l}_2 \ \ldots \ \mathbf{l}_M],$$

$$\mathbf{\Psi}\left(\mathbf{L}_x^T \tilde{\mathbf{X}}(1)\right) = \begin{bmatrix} \psi_1(\mathbf{l}_1^T \mathbf{x}_{t+\beta}) & \ldots & \psi_1(\mathbf{l}_1^T \mathbf{x}_{t+\beta+N-1}) \\ \psi_2(\mathbf{l}_2^T \mathbf{x}_{t+\beta}) & \ldots & \psi_2(\mathbf{l}_2^T \mathbf{x}_{t+\beta+N-1}) \\ \vdots & & \vdots \\ \psi_M(\mathbf{l}_M^T \mathbf{x}_{t+\beta}) & \ldots & \psi_M(\mathbf{l}_M^T \mathbf{x}_{t+\beta+N-1}) \end{bmatrix}.$$

4.3 *Nonlinear Function Estimation*

ALS and back-fitting approach in CANALS and PPR algorithms for estimating ϕ, θ, and ψ involve use of a regression technique capable of capturing nonlinear relationships. There are several regression techniques such as orthogonal polynomials, neural networks, local polynomial regression, smoothing splines, and kernel smoothers that may be grouped in terms of their common properties such as parametric/ nonparametric, fixed/adaptive, or local/global. No one technique is strictly superior to the others.

The modeling procedure requires a flexible, automated, and robust regression technique. Kernel regression, smoothing splines, and local polynomial regression techniques are flexible since they easily adapt to data because they are not constrained to any global parametric structure. Automating such techniques is a difficult task especially for serially correlated data. These techniques require the selection of some parameter value which determines the degree of smoothing such as bandwidth. Cross-validation is used frequently to determine the degree of smoothing, but for serially correlated data it may over-smooth or under-smooth. To avoid such difficulties, a fixed bandwidth is used throughout the regression.

These techniques are also sensitive to outliers. Robust techniques exist such as locally weighted scatter plot smoothing (LOWESS) (Cleveland, 1979). LOWESS iteratively smoothes with local polynomials. At each iteration, weights that are inversely proportional to the magnitude of residuals from the previous iteration are assigned to data and the regression is repeated. This greatly reduces sensitivity to outliers. In this work, assignment of weights to data and iterative smoothing are carried out by a supersmoother (Friedman, 1984) which is a local, adaptive, nonparametric regression technique. The functions ϕ, θ and ψ based on supersmoother estimates are local in the sense they are only defined within the domain of the data from which they are developed. Outside this domain, a linear relationship can be assumed. Let $\psi(x)$ represent the estimated supersmoother function where $l \leq x \leq u$. The final function including extrapolation is $\varphi(x)$

$$\varphi(x) = \begin{cases} a^l + b^l x & : \quad x < l \\ \psi(x) & : \quad l \leq x \leq u \\ a^u + b^u x & : \quad x > u \end{cases} \qquad (20)$$

Initial estimates of a^l, b^l, and a^u, b^u are found by regressing the lower and upper quartiles of observed $\psi(x)$ on x, respectively. The intercept terms a^l and a^u are then adjusted such that $a^l + b^l \min(x) = \psi(\min(x))$ and $a^u + b^u \max(x) = \psi(\max(x))$. Shifting of the intercepts allows for a smooth transition between domains. The functions estimated by the supersmoother are not considered continuous functions. Interpolation is done by regressing observed $\psi(x)$ on x using orthogonal Chebychev polynomials.

5. CSTR POLYMERIZATION MODELING

Data from a poly-vinyl acetate CSTR simulation (Teymour, 1989) is used to illustrate model identification. The outputs are reactor temperature (y_T) and number average molecular weight (MWn) of the polymer (y_M), and the manipulated input is residence time. The steady state gain of the system is not constant in the region of operation selected.

The input and outputs are sampled at 5 *min* intervals, and there is a time delay of 5 *min* for the input. The base input residence time levels are set at random from a uniform distribution between 10 and 90 *min* with a switching probability of 0.95. Added to this input is a signal with levels between -5 and 5. The range of input was chosen to exaggerate the nonlinearity of the system. Input switch levels are based on a uniform distribution with a switching probability of 0.95 and 0.80. 3000 samples of inputs and outputs are collected. To avoid numerical round off errors, MWn is divided by 1000 before model development. A known steady state operating point at a residence time of 50*min* was used to center the data. Gaussian random measurement noise was added to outputs prior to model development.

Latent Variable Model Identification. The latent variable model identification requires the specification of γ, β, and n. For this example, future and past horizons of 10 is used ($\gamma = \beta = 10$). The number of latent variables n is chosen by investigating the singular values of (14). For comparison, a linear model is also developed by linear CVA subspace identification.

Figure 1 shows the canonical correlation squared for linear and nonlinear models. The singular values for the linear and nonlinear model drop off significantly after 5 and 10 latent variables, respectively. The singular values can be interpreted as the canonical correlation between the future outputs and past inputs and outputs, after the effect of future inputs is removed. The canonical correlations of the nonlinear model are greater than the linear model. This is expected because the CANALS nonlinear CVA algorithm seeks to find nonlinear transformations of the original variables that maximize the canonical correlation. **Model Comparison**. A linear state space model

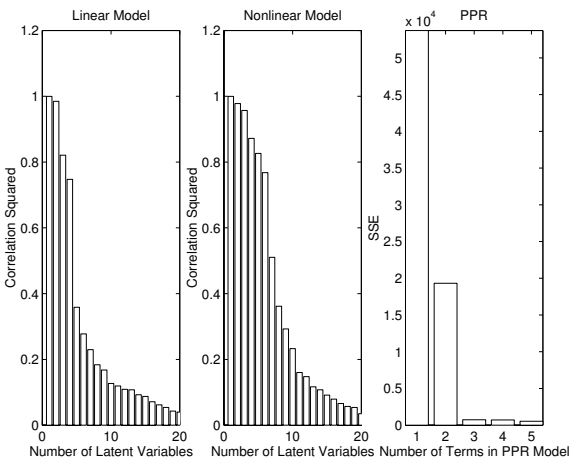


Fig. 1. Latent variable and PPR order selection.

found with CVA subspace modeling was compared to the nonlinear model with a linear or PPR link function. In-sample and out-of-sample prediction is evaluated by sum of squared error (SSE). Prediction consists of initializing the models from observed data then recursively simulating future outputs based on actual measured inputs and either past predicted states in the case of the state space model, or past predicted outputs in the case of the nonlinear model. To determine the form of the link function PPR was compared to linear least squares. The PPR model developed had 5 terms (M=5). The relative magnitudes of SSE in Figure 1 indicates that 3 terms are sufficient. A plot of the in-sample prediction of reactor temperature for various models (Figure 2) indicate that nonlinear models outperform the linear model. The SSE used for comparison is

$$SSE = \sum_t \left[\left(\frac{y_M(t) - \hat{y}_M(t)}{\sigma_M} \right)^2 + \left(\frac{y_T(t) - \hat{y}_T(t)}{\sigma_T} \right)^2 \right]$$

where y and $\hat{y}$ are the actual and predicted outputs respectively, and σ is the standard deviation of the observed variable. The SSE is 530 for the nonlinear model with linear link function, 558 for the nonlinear model with PPR link function, and 930 for the linear state space model. For the

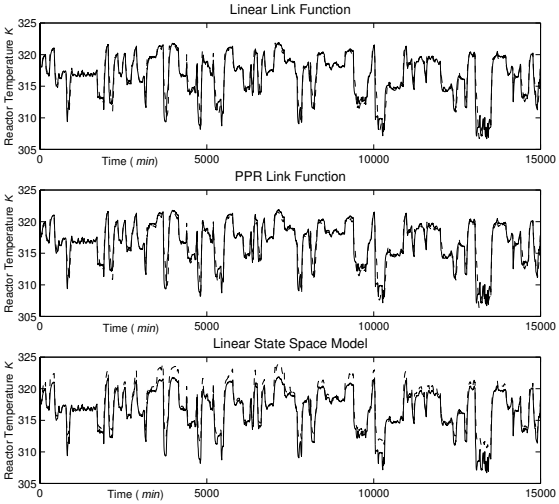


Fig. 2. Actual (-) and predicted (- -) reactor temperature. Top: nonlinear model with linear link; Center: nonlinear model with PPR link, and Bottom: linear model.

out-of-sample comparison 20 runs with different inputs and noise sequences were simulated in the same manner as the in-sample case. Nonlinear models clearly outperform the linear model with respect to SSE (Figure 3). The nonlinear models with linear and PPR link function have comparable performance with the linear link function model with slightly better performance. **Steady-State Analysis.** Numerical continuation is implemented with AUTO (Doedel *et al.*, 1998) to determine the steady-state characteristics of the

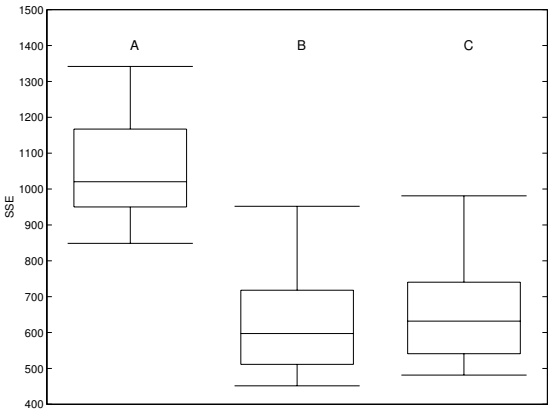


Fig. 3. SSE for 20 runs for (A) linear model, (B) nonlinear model with linear link, and (C) nonlinear model with PPR link.

empirical model. AUTO uses numerical continuation to trace out the fixed point solution given an initial steady state. The resulting steady-state curve is then compared to the steady-state curve of the physical model. For this comparison the linear link model was used. Let y_M^{ss} and y_T^{ss} be the steady state values of the outputs that correspond to the input u^{ss}. For the case of a least squares linear link function the model at steady state is

$$y_M^{ss} = \sum_{j=1}^{\beta} a_j^1 y_M^{ss} + \sum_{j=1}^{\beta} b_j^1 y_T^{ss} + \sum_{j=1}^{\beta} c_j^1 u^{ss}, \quad (21)$$

$$y_T^{ss} = \sum_{j=1}^{\beta} a_j^2 y_M^{ss} + \sum_{j=1}^{\beta} b_j^2 y_T^{ss} + \sum_{j=1}^{\beta} c_j^2 u^{ss}. \quad (22)$$

Figure 4 compares the actual and predicted fixed point steady state solutions. The nonlinear model predicts the fixed point solution inside the domain of the experimental data and the linear approximation works well outside.

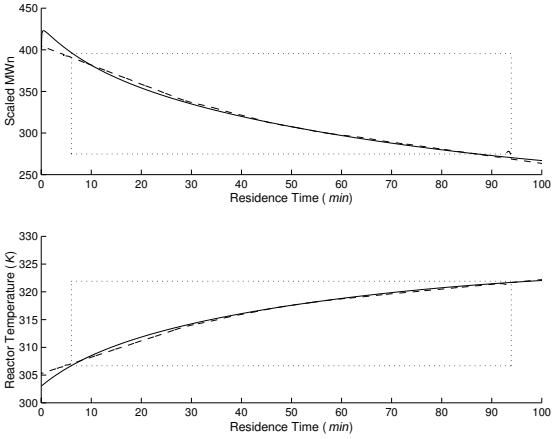


Fig. 4. Fixed point steady state solution of the actual physical model (-) and nonlinear empirical model (- -). The domain of experimental data is inside the dashed box (:).

6. CONCLUSION

The multivariate nonlinear empirical dynamic modeling technique is the extension of linear CVA subspace identification. A case study on modeling a polymerization in a CSTR illustrates the modeling approach and the dynamic and steady state performance of the nonlinear model which are better than the linear model performance.

7. REFERENCES

Breiman, L. and J. H. Friedman (1985). Estimating Optimal Transformations for Multiple Regression and Correlation. *J. Amer. Statist. Assoc.* **80**, 580–598.

Chen, R. and R. S. Tsay (1993). Nonlinear Additive ARX Models. *J. Amer. Statist. Assoc.* **88**(423), 955–967.

Cleveland, W.S. (1979). Locally-weighted Regression and Smoothing Scatterplots. *J. Amer. Stat. Assoc.* **74**, 829–836.

DeCicco, J. and A. Cinar (2000). Empirical Modeling of Systems with Output Multiplicities by Multivariate Additive NARX Models. *Indust. Eng. Chem. Research* **39**(6), 1747–1755.

Doedel, E. J., A. R. Champneys, T. F. Fairgrieve, Y. A. Kuznetsov, B. Sandstede and X. Wang (1998). AUTO 97: Continuation and Bifurcation Software for ODEs).

Friedman, J. and W. Stuetzel (1981). Projection Pursuit Regression. *J. Am. Stat. Assoc.* **76**, 817–823.

Friedman, J. H. (1984). A Variable Span Smoother. Technical Report No. 5. Dept. of Statistics, Stanford University.

Hastie, T. and R. Tibshirani (1990). *Generalized Additive Models.* Monographs on Statistics and Applied Probability. Chapman & Hall.

Jansson, M. and B. Wahlberg (1999). A Linear Regression Approach to State-Space Subspace System Identification. *Signal Processing* **52**(2), 103–129.

Larimore, W. (1990*a*). Identification and filtering of nonlinear systems using canonical variate analysis. In: *Nonlinear Modeling and Forecasting.* Addison-Wesley.

Larimore, W. (1990*b*). Canonical Variate Analysis in Identification, Filtering, and Adaptive Control. In: *Proc IEEE Conf. Decision Control.*

Teymour, F. (1989). The Dynamic Behavior of Free Radical Solution Polymerization Reactions in a Continuous Stirred Tank Reactor. PhD thesis. University of Wisconsin.

van der Burg, E. and J. de Leeuw (1983). Nonlinear Canonical Correlation. *British J. Math. Statist. Psychol.* **36**, 54–80.

Van Overschee, P. and B. De Moor (1994). N4SID: Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems. *Automatica* **30**(1), 75–93.

SEMI-BATCH TRAJECTORY CONTROL IN REDUCED DIMENSIONAL SPACES

Jesus Flores-Cerrillo and John F. MacGregor

Department of Chemical Engineering, McMaster University,
Hamilton, Ontario, Canada, L8S4L7

Abstract: A novel inferential strategy for controlling end-product quality properties using complete trajectories of manipulated variables is presented. Control through complete trajectory manipulation using empirical models only is possible by controlling the process in the reduce space (scores) of a latent variable model rather than in the real space of the manipulated variables. Model inversion and trajectory reconstruction is achieved by exploiting the correlation structure in the manipulated variable trajectories captured by a Partial Least Squares (PLS) model. The approach is illustrated with a condensation polymerisation example for the production of nylon. The data requirements for building the model are shown to be modest.

Keywords: Batch control, Partial Least Squares, Statistical process control, Condensation polymerisation, Reduced space control, Trajectory control.

1. INTRODUCTION

Batch/semi batch processes are commonly used because their flexibility to manage many different grades and types of products. In these processes, it is necessary to achieve tight final quality specifications. However, this is not easily achieved because batch operations suffer from constant changes in raw material properties, variations in start-up initialisation, and in operating conditions, all of which introduce disturbances in the final product quality. Moreover, compensating for these disturbances is difficult due to the non-linear behaviour of the chemical reactors and to the fact that robust on-line sensors for quality variable monitoring are rarely available.

Several approaches based on complex theoretical models and computationally intensive control strategies have been presented to control quality properties in batch processes (Kozub, 1989.) However, these strategies are difficult to implement because they require almost perfect model knowledge. Empirical modelling, on the other hand, has the advantage of using information routinely collected and of ease in model building. Yabuki and MacGregor, (1997) used empirical models for the control of product quality-properties. However, control action was restricted to only a few movements in the manipulated variables (injection of reactants) due to effective control action can only be applied at certain reaction stages.

In batch operation is not uncommon to find processes in which the quality properties must be controlled by adjusting several manipulated variables trajectories (MVT) through most of the duration of the process (for example, reactor temperature or pressure). In this case, the conventional approach is to coarsely segment the MVT into a few intervals or decision points (usually 5-10) and characterize them by slope and level (stair-case parameterisation, Russell et al., 1998). Therefore, in controlling a new batch, only the level and/or the slope of such intervals need to be adjusted because it is assumed that the MVT remains constant (same level/slope) until the next decision point. In this form, the number of parameters to be estimated from identification experiments remains relatively small. Studies involving this type of parameterisation can be found in Russell et al., (1998), and Lee, et al., (2001) among others. However, if fine trajectory segmentation is required

or if smoother MVTs need to be implemented, a much more comprehensive experimental design need to be performed to allow for an adequate identification of the effect of MV's on the controlled variables over the entire batch trajectory. Moreover, model inversion would be usually difficult because a large number of highly correlated control actions need to be determined at every decision point. A solution to this dilemma is to project such highly correlated process trajectories (MVT and measurements) into lower-dimensional spaces and to perform the control computation in the reduced dimension space. By projecting the original correlated trajectories into a lower dimension we are obtaining a few orthogonal variables that summarizes the original information. In this form, the model parameter estimation is more efficient and the control computation easier. In spite of the inherent advantages in controlling the MVT's of batch processes in the latent variable space, no literature has yet addressed this issue.

Statistical controllers for continuous processes based on Principal component analysis (PCA) have been proposed (Cheng and McAvoy, 1996; Chen et al., 1998), which also express the control objective in the score space of the PCA model. However, the approach taken here is different.

The purpose of this study is to introduce a novel inferential control strategy that allows a much finer characterization and smoother reconstruction (model inversion) of manipulated variable trajectories than those obtained using staircase parameterisation, without increasing the complexity and number of identification experiments needed for model building. These objectives are made possible by formulating the control strategy in the reduced dimensional space of a latent variable model, and then using the model to invert the solution for the MVT's. The contents of this work are as follow: in section 2 the methodology is introduced; in section 3, the control approach is illustrated with a condensation polymerisation case study for the production of nylon 6,6. In section 4, conclusions are drawn.

2. CONTROL METHODOLOGY

2.1 Model building

The proposed methodology uses historical-data bases and a few complementary identification experiments for model building. The empirical model is obtained using Partial Least Squares (PLS). However, other projection methods such as principal component regression may also be applied.

The database from which the PLS model is identified consists of a regressor matrix (**X**) composed of k row vectors ($\mathbf{x}^T$) of on-line process variable trajectories ($\mathbf{x_{on}}$) and possibly off-line measurements ($\mathbf{x_{off}}$), collected occasionally through the batch, $\mathbf{x_m^T} = [\mathbf{x_{on}^T} \quad \mathbf{x_{off}^T}]$, full manipulated variable trajectories (MVT) $\mathbf{u_c}$, and the matrix (**Y**) of quality properties measured at the end of the batch. Full MVT's are obtained through trajectory segmentation as illustrated in Figure 1. In this Figure is shown that the MVT's are finely segmented and that decision points (θ_i, i=1,2,…), where control action is taken, are chosen. Notice that the segment size is not necessarily uniform and that decisions points may be chosen arbitrarily. (However, the decision points will usually be selected using prior process knowledge.) In the limit, control action can be taken at every segment (i.e. every segment would represent a decision point).

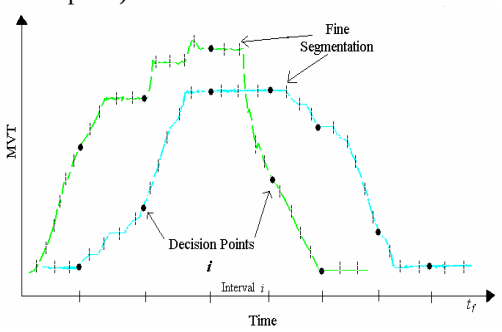


Fig. 1. Fine segmentation of MVT and decision points.

Linear PLS regression is performed by projecting the mean centered and scaled variables onto lower dimensional subspaces:

$$\mathbf{X} = \mathbf{TP}^T + \mathbf{E} \qquad (1)$$
$$\mathbf{Y} = \mathbf{TQ}^T + \mathbf{F}$$

where **T** are new latent variables **T=XW*** that capture most of the data variability, $\mathbf{P}^T$ is the loading matrix, and **E** and **F** are residual matrices. Non-linear PLS regression can also be used (Flores-Cerrillo, 2003). However, for simplicity, through this presentation linear models will be assumed.

The control methodology used in this work consists of two stages: 1) at predetermined decision times (θ_i, i=1,2,…) an inferential end-quality prediction using on-line and possible off-line process measurements ($\mathbf{x_m}$) and the MVT's ($\mathbf{u_c}$) up to the current time is performed to determine whether or not the controlled end-qualities (**y**) fall outside a non-control region, and if needed, 2) model inversion to obtain tne modified MVT for the remainder of the batch that will yield the desired final qualities. This two-stage procedure is repeated at every decision point (θ_i) using all available measurement and MVT's information up to that time. The novelty of the proposed approach is that the model inversion stage is performed in the reduced dimensional space (latent variable or score space) of a PLS model rather than in the real space of the MVT's. Due to the high correlation of measurements and control actions, the true dimensionality of the process, determined in the score variable space (t_a, a=1,2,…,A) of the PLS model, is generally much smaller than the number of manipulated variables points obtained from the MVT segmentation ($\mathbf{u_c}$). Therefore, the control computation performed in the reduced latent variable space (**t**) is much simpler than that performed in the real space. In the following the control methodology

is described for one control decision point during the batch. This is repeated at each future decision point.

2.2 Prediction

For on-line end-quality estimation ($\hat{\mathbf{y}}$), when a new batch *k* is being processed, at every decision point (θ_i, *i*=1,2,…) $0 \le \theta_i \le \theta_f$, there exists a regressor row vector $\mathbf{x}^T$ composed of, at least, the following variables

$$\mathbf{x}^T = [\mathbf{x}_m^T \quad \mathbf{u}_c^T] = \qquad (2)$$
$$[\mathbf{x}^T_{m,measured,q_i} \quad \mathbf{x}^T_{m,missing} \quad \mathbf{u}^T_{c,implemented,q_i} \quad \mathbf{u}^T_{c,future}]$$

The regressor vector **x** consists of 1) all measured variables ($\mathbf{x}_{m,measured}$) available up to time θ_i ($0 \le \theta \le \theta_i$), 2), unmeasured variables ($\mathbf{x}_{m,missing}$) not available at θ_i, but that will be available in the future ($\theta_{i+1} \le \theta \le \theta_f$), implemented control actions $\mathbf{u}_{c,implemented}$ ($0 \le \theta \le \theta_{i-1}$), and future control actions $\mathbf{u}_{c,future}$, ($\theta_i \le \theta \le \theta_f$) which will be determined through model inversion. Note that at the model building stage, the $\mathbf{x}_{m,missing}$ and $\mathbf{u}_{c,future}$ vectors are available for each batch.

To estimate whether or not the quality properties, for a new batch, will lie within an acceptable region, the prediction is performed considering $\mathbf{u}_{c,future} = \mathbf{u}_{c,nominal}$ (i.e. assuming that the remaining trajectory will be kept at their nominal conditions) using the PLS model:

$$\hat{\mathbf{t}}^T_{present} = [\mathbf{x}_m^T \quad \mathbf{u}_c^T]\mathbf{W}^* = \qquad (3)$$
$$[\mathbf{x}^T_{m,measured,q_i} \quad \mathbf{x}^T_{m,missing} \quad \mathbf{u}^T_{c,implemented,q_i} \quad \mathbf{u}^T_{c,n}]\mathbf{W}^*$$

$$\hat{\mathbf{y}}^T = \hat{\mathbf{t}}^T_{present}\mathbf{Q}^T \qquad (4)$$

$\mathbf{W}^*$ and $\mathbf{Q}^T$ are projection matrices obtained from the PLS model building stage (Geladi et al., 1986). $\hat{\mathbf{t}}_{present}$ is the projection of the **x** vector onto the reduced dimension space of the latent variable model (scores) at time θ_i, and $\hat{\mathbf{y}}$ is the vector of predicted end-quality properties. From the above equations, it can be noticed that changes in batch operation detected by process measurements ($\mathbf{x}_m$) or produced by changes in the MVT's ($\mathbf{u}_c$) would produce changes in the scores ($\hat{\mathbf{t}}_{present}$) and therefore in the end-quality properties (i.e. changes in the end-qualities can be detected through changes in the scores).

From equation (3), it can be noticed that in order to compute $\hat{\mathbf{t}}_{present}$ and $\hat{\mathbf{y}}$, it is necessary to have an estimate of the unknown future measurements ($\mathbf{x}_{m,missing}$) from ($\theta_{i+1} \le \theta \le \theta_f$). These can be obtained using efficient missing data algorithms available in the literature (Nelson et al., 1996). Alternatively, a multi-model approach in which a model is identified at every decision point can be used as discussed in Russell et al., (1998). The decision of one alternative over other depends on the number of decision points and/or performance of the missing data algorithm. In the example shown in this paper a single PLS model is used for control and the estimation of unknown future measurements is done by a missing data algorithm.

The non-control region can be determined in several ways, such as the one that takes into account the uncertainty of the model for prediction (Yabuki and MacGregor, 1997), from product specifications or from quality data under normal ("in-control") operating conditions. In this work a simple control region based on product quality specifications will be used (section 3).

If the quality prediction is outside the non-control region, then model inversion to obtain the MVT is needed. Obtaining of the full MVT consist of two stages: 1) Computation of the deviation of the scores from the quality targets and 2) Model inversion to obtain the real MVT using the correlation structure of the PLS model. These two stages are explained in as follows.

2.3 Control Computation

At every decision point (θ_i), the distance that the scores need to be changed ($\Delta\mathbf{t}$) to track the end-qualities closer to their set-points ($\mathbf{y}_{sp}$) can be obtained by solving the linear quadratic regulator (5):

$$\min_{\Delta\mathbf{t}(q_i)} (\hat{\mathbf{y}} - \mathbf{y}_{sp})^T \mathbf{Q}_1 (\hat{\mathbf{y}} - \mathbf{y}_{sp}) + \Delta\mathbf{t}^T \mathbf{Q}_2 \Delta\mathbf{t} + \lambda T^2$$
$$st \quad \hat{\mathbf{y}}^T = (\Delta\mathbf{t} + \hat{\mathbf{t}}_{present})^T \mathbf{Q}^T$$
$$T^2 = \sum_{a=1}^{A} \frac{(\Delta t + \hat{t}_{present})_a^2}{s_a^2} \qquad (5)$$
$$\Delta\mathbf{t}_{min} \le \Delta\mathbf{t} \le \Delta\mathbf{t}_{max}$$

where $\Delta\mathbf{t}^T = \mathbf{t}^T - \hat{\mathbf{t}}^T_{present}$, $\mathbf{Q}_1$ is a diagonal weighting matrix, $\mathbf{Q}_2$ is a movement suppression matrix, T^2 is the Hotelling's statistic, s_a^2 is the variance of the score t_a, and λ is a weighting factor. Hard constrains in the adjustment to the scores ($\Delta\mathbf{t}_{min} \le \Delta\mathbf{t} \le \Delta\mathbf{t}_{max}$) are problem dependent and may or not need to be included.

Equation (5) is a quadratic programming problem that can be restated as:

$$\min_{\Delta\mathbf{t}(q_i)} \frac{1}{2}\Delta\mathbf{t}^T\mathbf{H}\Delta\mathbf{t} + \mathbf{f}^T\Delta\mathbf{t} \qquad (6)$$

where

$$\mathbf{H} = \mathbf{Q}^T\mathbf{Q}_1\mathbf{Q} + \mathbf{Q}_2 + \mathbf{Q}_3$$
$$\mathbf{f}^T = (\mathbf{Q}\hat{\mathbf{t}}_{present})^T\mathbf{Q}_1\mathbf{Q} + \hat{\mathbf{t}}^T_{present}\mathbf{Q}_3 \qquad (7)$$
$$\mathbf{Q}_3 = \mathrm{diag}\left[\lambda / s_a^2\right]$$

and whose interpretation is given in Figure 2 for a two dimensional space. As can be seen in this Figure, the aim of equations (6-7) is to reduce the distance of $\hat{\mathbf{t}}_{present}$, by an amount D**t**, to get closer to the score value corresponding to the quality set-points

($\mathbf{t}_{sp} = (\mathbf{Q}^T\mathbf{Q})^{-1}\mathbf{Q}^T\mathbf{y}_{sp}$). Due to the movement suppression matrix ($\mathbf{Q_2}$) and/or λ, the achieved **t** may not achieve $\mathbf{t}_{sp}$, but will be closer to it. If we desire to obtain the D**t** that would force the calculated $\mathbf{t}^T = \mathbf{t}_{sp}^T$, we could use a minimum-variance like controller. Under this situation equation (5), with $\mathbf{Q_2}=\mathbf{I}$ and λ=0, can be restated as:

$$\min_{\Delta\mathbf{t}(q_i)} \Delta\mathbf{t}^T\Delta\mathbf{t} \quad st \quad \mathbf{y}_{sp}{}^T = (\Delta\mathbf{t} + \hat{\mathbf{t}}_{present})^T\mathbf{Q}^T \tag{8}$$

and whose solution can be easily obtained as:

$$\Delta\mathbf{t}^T = (\mathbf{y}_{sp}{}^T - \hat{\mathbf{t}}_{present}^T\mathbf{Q}^T)(\mathbf{Q}\mathbf{Q}^T)^{-1}\mathbf{Q} = \mathbf{t}_{sp}^T - \hat{\mathbf{t}}_{present}^T\mathbf{Q}^T(\mathbf{Q}\mathbf{Q}^T)\mathbf{Q} \tag{9}$$

If we consider **y** to be deviations from $\mathbf{y}_{sp}$, then $\mathbf{t}_{sp}=\mathbf{0}$ and the last equation is reduced to:

$$\Delta\mathbf{t}^T = -(\hat{\mathbf{t}}_{present}^T\mathbf{Q}^T)(\mathbf{Q}\mathbf{Q}^T)^{-1}\mathbf{Q} \tag{10}$$

A detuning factor (d) may be included for this minimum-variance like controller to achieve some robustness against model error:

$$\Delta\mathbf{t}^T = -\boldsymbol{\delta}(\hat{\mathbf{t}}_{present}^T\mathbf{Q}^T)(\mathbf{Q}\mathbf{Q}^T)^{-1}\mathbf{Q} \tag{11}$$

where $\mathbf{0}\le d\le\mathbf{1}$. Δ**t** is computed at every decision point (θ_i).

Notice that the matrix $\mathbf{QQ}^T$ has dimension $m \times m$ (m being the number of quality properties). Therefore, in order to do not have an ill-conditioned matrix inversion, the quality properties should not be highly correlated. This poses no problem since one can always perform a PCA on the **Y** quality matrix to obtain a set of orthogonal variables (t) that can be used as new controlled variables, or perform selective PCA (Jackle and MacGregor, 1998) on the **Y** matrix to determine the best independent subset of quality variables to be controlled. Removal of a high correlated *y* variable should not be detrimental to its control since, by controlling the other quality variables, that quality variable will also be controlled.

2.4 Inversion of PLS model to obtain the MVT's

Once the low dimensional ($1\times A$) vector D**t** is computed via one of the control algorithms in the last section, it remains to reconstruct from it, the high dimensional trajectories for the future process variables ($\mathbf{x}_{m,missing}$) and for the future manipulated variables ($\mathbf{u}_{c,future}$) over the remainder of the batch. These future trajectories can be computed from the PLS model (1) in such a way that their covariance structure is consistent with past operation. If there were no additional restrictions on the trajectories, such as might exits for a control action at θ=0, then the model for the X-space can be used directly to compute the **x** vector trajectory for the entire batch (Jaeckle and MacGregor, (1998)) as:

$$\Delta\mathbf{x}^T = [\Delta\mathbf{t}^T]\mathbf{P}^T \tag{12}$$

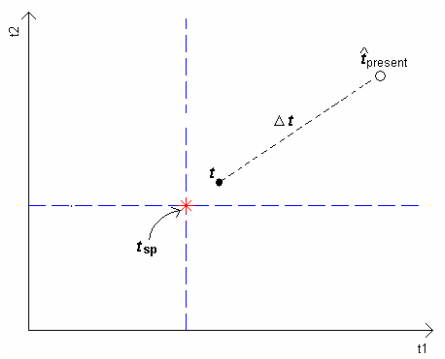


Fig. 2. Control in the reduce space (score control).

However for control intervals at times θ_i >0, there already exits observed trajectories for the interval $0 \le \theta < \theta_i$, for the measured process variables ($\mathbf{x}_{m,measured}$) and for the already implemented manipulated variables ($\mathbf{u}_{m,implemented}$) that must be respected when computing their trajectories for the remainder of the batch ($\theta_i \le \theta \le \theta_f$). From equation (3) it can be seen that the changes in the score vector, D**t**, is related to the changes in the nominal trajectories according to:

$$\Delta\mathbf{t}^T = [\Delta\mathbf{x}^T]\mathbf{W}^* = [\Delta\mathbf{x}^T_{m,measured}\ \Delta\mathbf{u}^T_{c,implemented}\ \Delta\mathbf{x}^T_{m,missing}\ \Delta\mathbf{u}^T_{c,future}]\mathbf{W}^* \tag{13}$$

If one is currently at decision time θ_i, then clearly $\Delta\mathbf{x}_{m,measured} = \mathbf{0}$ and $\Delta\mathbf{u}_{m,implemented} = \mathbf{0}$, and the remaining trajectories to be computed for $\theta_i \le \theta \le \theta_f$ (i.e. $\Delta\mathbf{x}_{m,missing}$ and $\Delta\mathbf{u}_{c,future}$) should satisfy the following relation:

$$\Delta\mathbf{t}^T = [\mathbf{0}^T\ \ \mathbf{0}^T\ \ \Delta\mathbf{x}^T_{m,missing}\ \ \Delta\mathbf{u}^T_{c,future}]\mathbf{W}^* = [\mathbf{0}^T\ \ \Delta\mathbf{x}_2^T]\begin{bmatrix}\mathbf{W}_1^* \\ \mathbf{W}_2^*\end{bmatrix} \tag{14}$$

where $\Delta\mathbf{x}_2^T = [\Delta\mathbf{x}^T_{m,missing}\ \ \Delta\mathbf{u}^T_{c,future}]$ is the vector representing the change in future measurements and remaining MVT ($\theta_i \le \theta \le \theta_f$), and $\mathbf{W}_2^*$ its corresponding projection matrix. Then,

$$\Delta\mathbf{t}^T = [\Delta\mathbf{x}_2^T]\mathbf{W}_2^* \tag{15}$$

Furthermore, in order for the MVT and missing values to keep their correlation structure according to the PLS model (equation 12) the following condition must hold:

$$\Delta\mathbf{x}_2^T = \boldsymbol{\alpha}^T\mathbf{P}_2^T \tag{16}$$

This ensures that the relationship among all the process and manipulated variables trajectories that are being computed, will respect the nature of those trajectories in the data used to build the PLS model.

a can be estimated by substituting (16) in (15) according to:

$$\Delta\mathbf{t}^T = (\boldsymbol{\alpha}^T\mathbf{P}_2^T)\mathbf{W}_2^*$$
$$\boldsymbol{\alpha}^T = \Delta\mathbf{t}^T(\mathbf{P}_2^T\mathbf{W}_2^*)^{-1} \tag{17}$$

and by substituting (17) in (16), the MVT are obtained ($\theta_i \le \theta \le \theta_f$):

$$\Delta\mathbf{x}_2^T = \Delta\mathbf{t}^T(\mathbf{P}_2^T\mathbf{W}_2^*)^{-1}\mathbf{P}_2^T \tag{18}$$

It is easy shown that this reduces to the relationship in (12) when $\theta_i = 0$ where there are no previous trajectory measurements or manipulated variables.

The final control algorithm, in the case of linear models and no constrains, is obtained by substituting (11) in (18):

$$\Delta \mathbf{x}_2^T = -\delta(\hat{\mathbf{t}}_{\mathbf{present}}^T \mathbf{Q}^T)(\mathbf{Q}\mathbf{Q}^T)^{-1}\mathbf{Q}(\mathbf{P}_2^T\mathbf{W}_2^*)^{-1}\mathbf{P}_2^T \quad (19)$$

This inferential algorithm (19) is then repeated at every decision point (θ_i) until completion of the batch. Inversion of the $(A \times A)$ matrix $\mathbf{P}^T\mathbf{W}^*$ is nearly always well conditioned.

3. SIMULATION STUDIES

In the batch condensation polymerisation of nylon 6,6 the end product properties are affected by disturbances in the water content of the feed. In plant operation feed water content disturbances occurs because a single evaporator usually feeds several reactors (Russell et al., 1998). The non-linear model used in this work for data generation and model performance evaluation was developed by Russell et al., (1998). For a complete description of the model, and model parameters the reader is referred to the original publication.

Russell et al., (1998) studied this system and proposed several control strategies including conventional control (PID and gain schedule PID), non-linear model based control and empirical control based in linear state-space models. In their data-base approach, control of the system is achieved by reactor and jacket pressure manipulation. These two manipulated variables were segmented and characterised by slope and level (stair-case parameterisation) leading to 10 control variables. A total of 7 intervals (decision points) were used. The empirical state space model was identified from 69 batches arising from an experimental design. Several differences between the control strategy used by Russell et al., and the one proposed here can be noticed, the two most important being that: (*i*) the control is computed in the reduce latent variable space rather than in the real space of the MVT's, and that (*ii*) a much finer MVT reconstruction is achieved without increasing the complexity and number of experiments to be used in model building.

Control objectives and Trajectory segmentation.
The control objective is to obtain nylon 6,6 with an end-amine concentration (NH_2) of 49.33 and number averaged molecular weight (MWN) of 13533 (total reaction time 200 min), when the system is affected by changes in the initial water content (W). The MVT's used to control the end-qualities are the jacket and reactor pressure trajectories. These trajectories are finely segmented every 5 min. starting at 35 min. (of the beginning of the reaction) until 30 min. before the completion of the batch, giving a total of 40 control variables. Two control decision points at 38 and 75 min. were found to be necessary to yield adequate control for the conditions used in this example. In order to predict NH_2 and MWN, on-line measurements of the reactor temperature (Tr) and venting (v) are considered available every two minutes.

Data Generation: In the example that follows, a PLS model with 5 latent variables (determined by cross-validation) was built from a data set consisting of 15 batches in which *W* was randomly varied and 30 batches in which some movement in the MVT (at the two decision points) was performed (some of this data set may be available from historical data). However, adequate control performance has been achieved using only a total of 15 batches (Flores-Cerrillo, 2003).

3.1 Results

To illustrate the control performance of the algorithm, some results are presented. The first step is to determine if the prediction of the PLS model at the decision points is adequate. In Figure 3, the final qualities are shown for the case in which the water content randomly varies for 15 batches in the range of ±10%. The end quality property prediction should be performed at every decision point to determine if the next control action should be implemented or not. In Figure 3 prediction results at 38 min are shown. As can be seen in this figure, the predicted quality properties (□) using the PLS model are in good agreement with the observed values (o). Slight improvement in the predictions at high MWN and NH_2 values could be obtained with a non-linear PLS model (Flores-Cerrillo, 2003). However, the linear PLS model is very good in the target region (mid-values) and adequate in the extremes.

In Figure 4 is shown the performance of the controller algorithm (equation 19 with d=1.0) to the end-properties when the process is affected by the disturbances in the initial water concentration discussed above. In this Figure (o) represents what would happen if control action were not taken and (□) the qualities obtained after control is performed. As can be seen in this Figure, the proposed control scheme tracks on target all bad batches (inside the dotted box of quality specifications). Figure 5a and 5b show the jacket and reactor pressure MVT respectively for runs 1 and 15 together with their nominal conditions.

4. CONCLUSIONS

A novel control strategy for final product quality control in batch and semi-batch processes is proposed. The strategy recomputes on-line the entire remaining trajectories for the MV's at several decision points. However, in spite of the fact that the resulting controller consist of high dimensional manipulated variable trajectories (MVT's) the control algorithm involves only the solution for a small number of latent variables in the reduced dimensional space of a PLS model. The strategy uses empirical PLS models identified from historical data and a few complementary experiments. The strategy is illustrated using a simulated condensation

polymerisation process. Since smooth and continuous MV trajectories can be obtained, the approach seems well suited for use in processes and mechanical systems (robotics) where such smooth changes in the MV's are desirable.

5. ACKNOWLEDGMENTS

J. Flores-Cerrillo thanks McMaster University and SEP for financial support and to Dr. Russell S. A. for kindly providing us with his condensation polymerisation simulator.

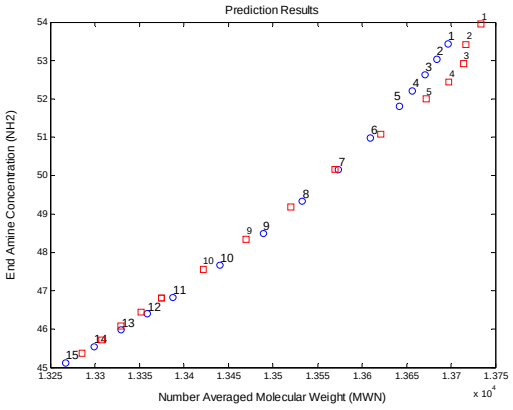


Fig. 3. Observed (o) and predicted (□) end-quality properties using PLS model.

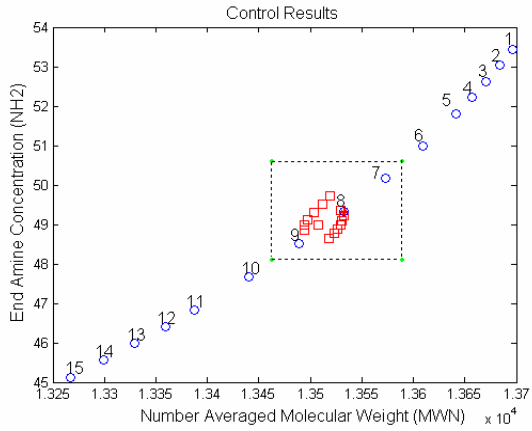
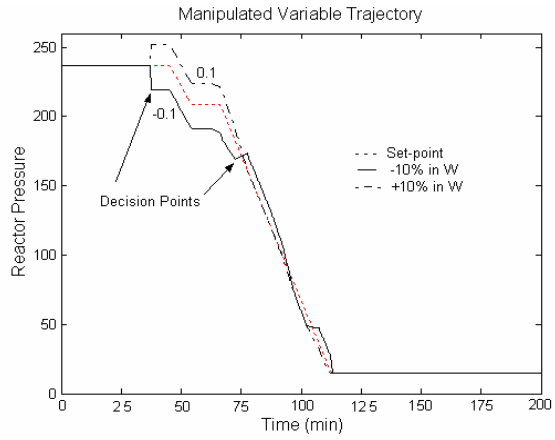
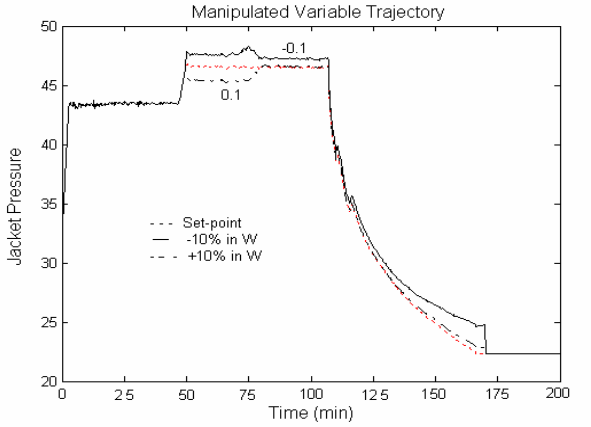


Fig. 4. Control results. (o) End-quality properties without control and (□) after control is taken.



a)



b)

Fig. 5a, 5b. Manipulated Variable Trajectories. (- - -) set-point, (—) when the disturbance is –10% in W, and (- – -) when disturbance is +10% in W. Reaction time 200min.

6. REFERENCES

Flores-Cerrillo, J. Ph. D. Thesis, McMaster University, Hamilton, Ontario, Canada, 2003 (expected graduation).

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

Jaeckle, J.M., and MacGregor, J.F. (1998). Product Design Through Multivariate Statistical Analysis of Process Data. *AIChE Journal*, ***44***, 1105-1118.

Kozub, D. J. (1989). Multivariate Control Design – Robustness and Nonlinear Inferential Control for Semi Batch Polymerization Reactors. Ph. D. Thesis, Department of Chemical Engineering, McMaster University, Canada.

Lee, J., Lee, K. S., J. H. Lee. and P. Sunwon (2001). An on-line batch span minimization and quality control strategy for batch and semi-batch processes. *Control Engineering Practice*, **9**, 901-909.

Nelson, P., MacGregor, J. F. and P.A. Taylor (1996). Missing Data Methods in PCA and PLS: Score Calculations with Incomplete Observations. *Chemometrics and Intelligent Laboratory Systems*, **35**, 45-65.

Russell, S. A., P. Kesavan., J. H. Lee. and B. A. Ogunnaike (1998). Recursive Data-Based Prediction and Control of Batch Product Quality. *AIChE Journal*, **44**, 2442-2458.

Yabuki, Y. and J.F. MacGregor (1997). Product Quality Control in Semibatch Reactors Using Midcourse Correction Policies. *Ind. Eng. Chem. Res.*, **36**, 1268-1275.

Chen, G. and McAvoy, T. J. (1996). Process control utilizing data based multivariate statistical models. *The Canadian Journal of Chemical Engineering*, ***74***, 1010-1024.

Chen, G., McAvoy, T. J., and Pivoso, M. J. (1998). A multivariate statistical controller for on-line quality improvement. *J. Proc. Cont.*, ***2***, 139-149.

MODIFIED SUBSPACE IDENTIFICATION METHOD FOR BUILDING A LONG-RANGE PREDICTION MODEL FOR INFERENTIAL CONTROL

Yangdong Pan, Jay H. Lee [1]

School of Chemical Engineering, Georgia Institute of Technology, Atlanta, GA

Abstract: In a chemical plant involving a series of processing units, it is beneficial to have a model that can accurately forecast the behavior of downstream variables based on upstream measurements. Such a model can be useful in feedforward and inferential control of the downstream variables to compensate for various upstream disturbances. However, creating such a dynamic model can be very difficult. The conventional multivariable identification approach based on minimizing single-step-ahead prediction error, can result in models leading to poor prediction and control in the described context. To alleviate this difficulty, we propose a modification to the conventional subspace identification method geared towards accurate k-step-ahead prediction, where k is a number chosen according to the estimated dead time. It is shown that the modified subspace identification method can be used in conjunction with the k-step prediction error minimization (PEM). Using an illustrative examples involving six mixing units with a recycle loop, we demonstrate the improvement that is possible from adopting the suggested modification.

1. INTRODUCTION

Most modern plants involve a large number of interconnected processing units, thus raising the need to consider the interactions and information flows among them. A typical plant setup involves measurements and manipulated variables located at the upstream and downstream property variables that need to be controlled. For disturbances occurring in the feed or upstream units, the upstream variables show more immediate responses. Their quick responses, if measured, can be used to manipulate upstream processing conditions in order to keep the downstream properties in control – as in feedforward control or inferential control. To realize this, the upstream measured process variables must be accurately related to the downstream property variables in a dynamic manner. The same situation appears in distributed parameter systems with a large residence time, such as a continuous pulp digester.

Developing a model that accurately captures the dynamic correlation between upstream and downstream variables presents a major challenge. Such models are likely to involve large time delays and dynamics of high order and possibly multiple time scales (due to recycle loops commonly found in industrial plants). Any one of the above features can pose difficulties for the existing system identification approaches. Furthermore, inferential control puts a higher demand on the model accuracy.

In the described problem's context, it is obvious that long-range prediction performance of the model is what ultimately matters. Since a large dead time is involved typically, the short-term predictions, however accurate they may be, are not useful. The importance of emphasizing the long-range prediction over the short-term prediction becomes more clear when one considers the significant model bias typical in most system identification carried out in practice. In the literature, the minimization of k-step-ahead prediction error in the prediction error minimization (PEM) method has been suggested and discussed [8][10] . In addition to the time-domain interpretation, Wahlberg and Ljung [6] formally showed that the

[1] To whom all correspondence should be addressed. phone (404)385-2148, fax (404)894-2866, e-mail:Jay.Lee@che.gatech.edu

use of k-step-ahead prediction methods amounts to emphasizing the accuracy of low-frequency dynamics more in distributing the bias, compared to the conventional one-step-ahead error minimization, which tends to put higher emphasis on the high frequency behavior.

In spite of these developments, understanding of where and how to use the more general k-step PEM in process control's context has been fairly limited. The few exceptions include papers by Shook *et al* [2], and Huang *et al* [3]. Still, a clear link between the method and situations or types of process applications, from which substantial benefits of the method are likely to be realized, is not there. Another reason for the lack of its use in practice is the numerical difficulty associated with using k-PEM for multivariable systems. In addition to the usual complexities (*e.g.*, local minima) associated with the standard PEM, the design of the prefilter necessary to turn the multi-step-ahead prediction error minimization into the one-step-ahead prediction error minimization requires the noise model, which is usually not known *a priori*. In many works, such as the long-range predictive identification (LRPI) approach advocated by Shook *et al* [2], the noise model is assumed to be fixed *a priori*. In this case, the quality of the identified model as well as the performance of the final predictive controller can be strongly influenced by the choice of the noise model.

For multivariable identification problems, the subspace identification method has many attractive features, including the numerical robustness and non-iterative nature of the algorithm [9]. However, the conventional subspace identification method is geared implicitly towards providing accurate one-step-ahead predictions. It is shown in this paper that, for those applications requiring accurate long-range predictions, the conventional method can perform poorly. Given the above-mentioned merits of the subspace method, however, it is useful to consider how the method can be extended to give higher emphasis on the long-range prediction performance.

The contribution of this paper can be two-fold. First, we bring to attention a situation ubiquitous in the process industries, for which the importance of fitting a model to optimize its long-range prediction performance is very high. Second, we present a modified version of subspace identification, in which the emphasis is given to the k-step-ahead prediction performance, where k is a general number chosen according to the process dead-time. We also show how a model obtained from the modified subspace method can be further improved through the k-step-ahead prediction error minimization (k-PEM). An example involving 6 mixing units with a recycle loop is chosen to show the importance of emphasizing the long-range performance through the proposed method.

2. PROPOSED MODIFICATIONS FOR EMPHASIZING THE K-STEP-AHEAD PREDICTION PERFORMANCE

Here we propose a modification to the conventional identification method with the aim of obtaining more accurate k-step-ahead predictions. We first show the modifications for the subspace identification method. After that, we discuss how the resulting model can be improved through the PEM method.

2.1 *Subspace Identification Based on Minimizing the k-Step-Ahead Prediction Error*

The conventional subspace ID approach, such as the N4SID method described in [9], implicitly assumes that the purpose of the model is to provide accurate one-step-ahead prediction. This is seen in the step where state space matrices A, B, C, D are estimated through least squares. In N4SID, data bank for one-step ahead Kalman state estimate $x_{t+1|t}$ is first created from the input/output data based on the following multi-step prediction equation:

$$\begin{bmatrix} y_{t+1} \\ y_{t+2} \\ \vdots \\ y_{t+\bar{n}} \end{bmatrix} = L_1 \begin{bmatrix} y_{t-\bar{n}+1} \\ y_{t-\bar{n}+2} \\ \vdots \\ y_t \end{bmatrix} + L_2 \begin{bmatrix} u_{t-\bar{n}+1} \\ u_{t-\bar{n}+2} \\ \vdots \\ u_t \end{bmatrix} + L_3 \begin{bmatrix} u_{t+1} \\ u_{t+2} \\ \vdots \\ u_{t+\bar{n}-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_{t+1|t} \\ \varepsilon_{t+2|t} \\ \vdots \\ \varepsilon_{t+\bar{n}+1|t} \end{bmatrix} \quad (1)$$

Since we can write the optimal predictions in terms of the Kalman state estimate (i.e., the estimate by the nonstationary Kalman Filterinitialized at $t - \bar{n} + 1$ as

$$\begin{bmatrix} y_{t+1|t} \\ y_{t+2|t} \\ \vdots \\ y_{t+\bar{n}|t} \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{\bar{n}-1} \end{bmatrix} x_{t+1|t} + L_3 \begin{bmatrix} u_{t+1} \\ u_{t+2} \\ \vdots \\ u_{t+\bar{n}+1} \end{bmatrix}, \quad (2)$$

$$\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{\bar{n}-1} \end{bmatrix} x_{t+1|t} = \begin{bmatrix} L_1 & L_2 \end{bmatrix} \begin{bmatrix} y_{t-\bar{n}+1} \\ \vdots \\ y_t \\ u_{t-\bar{n}+1} \\ \vdots \\ u_t \end{bmatrix} \quad (3)$$

Because state coordinates are not fixed *a priori*, one-step ahead state estimate $x_{t+1|t}$ can be created by estimating $[L_1 \quad L_2]$ through least squares

and then finding a set of basis that spans its range space. In N4SID, this is done through a series of oblique matrix projections [9]. Once data for $x_{t+1|t}$ and $x_{t+2|t+1}$ are created, the state space matrices are obtained by solving the linear least squares problem

$$\begin{aligned} x_{t+2|t+1} &= Ax_{t+1|t} + Bu_{t+1} + w_{t+1|t} \\ y_{t+1} &= Cx_{t+1|t} + \varepsilon_{t+1|t} \end{aligned} \tag{4}$$

where the residuals w and ε are minimized. Hence, in this step of the subspace method, one-step-ahead prediction error is minimized. The covariance matrix for w and ε, $\begin{pmatrix} R_w & R_{w,\varepsilon} \\ R_{w,\varepsilon}^T & R_\varepsilon \end{pmatrix}$, is estimated from the residuals of the least squares and the Kalman filter is designed with the calculated system and covariance matrices to obtain the following innovation form of the model.

$$\begin{aligned} x_{t+2|t+1} &= Ax_{t+1|t} + Bu_{t+1} + K\varepsilon_{t+1|t} \\ y_{t+1} &= Cx_{t+1|t} + \varepsilon_{t+1|t} \end{aligned} \tag{5}$$

We may generalize N4SID to emphasize the k-step-ahead prediction in the following manner. To create k-step ahead state estimates, the optimal multi-step prediction equation of (1) can be modified to

$$\begin{aligned} \begin{bmatrix} y_{t+k} \\ y_{t+k+1} \\ \vdots \\ y_{t+k+\bar{n}} \end{bmatrix} &= L_1 \begin{bmatrix} y_{t-\bar{n}+1} \\ y_{t-\bar{n}+2} \\ \vdots \\ y_t \end{bmatrix} + L_2 \begin{bmatrix} u_{t-\bar{n}+1} \\ u_{t-\bar{n}+2} \\ \vdots \\ u_{t+k-1} \end{bmatrix} \\ &\quad + L_3 \begin{bmatrix} u_{t+k} \\ u_{t+k+2} \\ \vdots \\ u_{t+k+\bar{n}-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_{t+k|t} \\ \varepsilon_{t+k+1|t} \\ \vdots \\ \varepsilon_{t+k+\bar{n}|t} \end{bmatrix} \end{aligned} \tag{6}$$

As before, it follows that

$$\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{\bar{n}-1} \end{bmatrix} x_{t+k|t} = \begin{bmatrix} L_1 & L_2 \end{bmatrix} \begin{bmatrix} y_{t-\bar{n}+1} \\ \vdots \\ y_t \\ u_{t-\bar{n}+1} \\ \vdots \\ u_{t+k-1} \end{bmatrix} \tag{7}$$

Following the same procedure as before, data bank for k-step-ahead state estimates $x_{t+k|t}$ and $x_{t+k+1|t+1}$ can be obtained. Then, a state space model can be obtained by performing least squares on the following equations:

$$\begin{aligned} x_{t+k+1|t+1} &= Ax_{t+k|t} + Bu_{t+k} + w_{t+k|t} \\ y_{t+k} &= Cx_{t+k|t} + \varepsilon_{t+k|t} \end{aligned} \tag{8}$$

The residual $\varepsilon_{t+k|t}$ represents the k-step-ahead prediction error, which is minimized. Note that, if the data-based Kalman estimates were perfect, then

$$w_{t+k|t} = \underbrace{A^{k-1}K}_{\bar{K}} \varepsilon_{t+1|t} \tag{9}$$

Also,

$$\varepsilon_{t+k|t} = \underbrace{\sum_{i=0}^{k-1} q^{-i} \mathcal{H}_i}_{\bar{F}_k(q)} \varepsilon_{t+k|t+k-1} \tag{10}$$

where $\mathcal{H}_i$ is the i^{th} Markov parameter of the noise model (A, K, C, I).

Based on these, the procedure for extracting $(A, B, C,)$ and K are as follows:

(1) Solve the least squares problem for the output equation to find C that minimizes $y_{t+k} - Cx_{t+k|t}$ in the 2-norm sense. The residuals represent the data for $\varepsilon_{t+k|t}$.
(2) Solve the least squares for the state equation to find A, B. The residual can be viewed as $w_{t+k|t}$.
(3) On the generated residual of $\varepsilon_{t+k|t}$, use a whitening filter to obtain one-step-ahead prediction error $\varepsilon_{t+k|t+k-1}$. A convenient way to do this is to apply subspace identification to the data. The ouput residual from this will be $\varepsilon_{t+1|t}$.
(4) Calculate the covariance matrix for $w(t+k|t)$ and using the whitened residual $\varepsilon(t+1|t)$. According to (9), the covariance matrix for the residual $w_{t+k|t}$ and $\varepsilon_{t+1|t)}$ has the form of

$$\begin{bmatrix} A^{k-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} R_w & R_{w,\varepsilon} \\ R_{w,\varepsilon}^T & R_\varepsilon \end{bmatrix} \begin{bmatrix} A^{k-1} & 0 \\ 0 & I \end{bmatrix}^T \tag{11}$$

where $\begin{pmatrix} R_w & R_{w,\varepsilon} \\ R_{w,\varepsilon}^T & R_\varepsilon \end{pmatrix}$ represents the covariance for $w_{t+1|t}$ and $\varepsilon_{t+1|t}$. With the calculated system matrices and the extracted covariance matrix for w and ε, one can proceed to design the Kalman filter to put the model in the innovation form. The k-step ahead predictor can be easily derived from it.

It should be obvious to those familiar with the subspace identification method that the asymptotic properties of N4SID such as unbiasedness and consistency remain intact with the above modifications.

2.2 *k-Step Prediction Error Minimization*

Although the modified subspace ID method puts higher emphasis on the accuracy of the k-step ahead prediction in obtaining state space matrices, it does not directly minimize k-step-ahead

prediction error for a finite data set. It has been suggested in Ljung [1] that the subspace method be used to initialize PEM, which generally requires a special parameterization and a good initial guess to be successful. Here we propose to use the model from the proposed k-step subspace ID method to start the k-step PEM.

A MIMO state space model,

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + Ke_t \\ y_t &= Cx_t + Du_t + e_t \end{aligned} \tag{12}$$

can be represented in the following input/output form:

$$y_t = G(q)u_t + H(q)e_t \tag{13}$$

where

$$\begin{aligned} G &= C(qI - A)^{-1}B + D \\ H &= C(qI - A)^{-1}K + I \end{aligned} \tag{14}$$

The optimal one-step ahead predictor is given by Ljung [1]

$$\hat{y}_{t|t-1} = H^{-1}Gu_t + (1 - H^{-1})y_t \tag{15}$$

If parameterized models G_θ and H_θ are used, then the optimal one-step-ahead predictor can be written as

$$\hat{y}_{t|t-1} = H_\theta^{-1}G_\theta u_t + (1 - H_\theta^{-1})y_t \tag{16}$$

Optimal k-step-ahead predictor is

$$\hat{y}_{t|t-k} = W_k G_\theta u_t + (1 - W_k)y_t \tag{17}$$

where

$$W_k = F_k H_\theta^{-1} \tag{18}$$

and

$$F_k = \sum_{i=0}^{k-1} \mathcal{H}_i q^{-i} \tag{19}$$

Here, $\mathcal{H}_i$ is a $n_y \times n_y$ matrix representing the i^{th} impulse response coefficient matrix of $H(q)$. The optimal k-step ahead predictor can also be viewed as the optimal one-step ahead predictor associated with the model

$$y_t = Gu_t + HF_k^{-1}\varepsilon_t \tag{20}$$

where ε_t is a white noise.

For a SISO system, F_k, if known, can be regarded as a prefilter and the k-step prediction error minimization is the same as the one-step prediction error minimization with the filtered I/O data. However, for a MIMO system, because matrices do not commute in multiplication, prefiltering the data before applying the one-step ahead PEM does not work. Therefore, F_k has to be embedded into the model structure when applying the PEM, resulting in a structured identification problem. Let us use the state-space representation of

$$\begin{aligned} F_k &= (\tilde{A}_F, \tilde{B}_G, \tilde{C}_G, \tilde{D}_G) \\ G &= (A, B, C, D) \\ H &= (A, K, C, I) \end{aligned} \tag{21}$$

First, the inverse system F_k^{-1} is,

$$\begin{aligned} F_k^{-1} = (&\tilde{A}_F - \tilde{B}_F\tilde{D}_F^{-1}\tilde{C}_F, \tilde{B}_F\tilde{D}_F^{-1}, \\ &-\tilde{D}_F^{-1}\tilde{C}_F, \tilde{D}_F^{-1}) \end{aligned} \tag{22}$$

Let us denote

$$F_k^{-1} = (\tilde{A}_{F^{-1}}, \tilde{B}_{F^{-1}}, \tilde{C}_{F^{-1}}, \tilde{D}_{F^{-1}}) \tag{23}$$

where $\tilde{D}_{F^{-1}} = I$. Then, the combined model structure HF_k^{-1} is,

$$\begin{aligned} &HF_k^{-1} = \\ &\left(\begin{bmatrix} A & K\tilde{C}_{F^{-1}} \\ 0 & \tilde{A}_{F^{-1}} \end{bmatrix}, \begin{bmatrix} K\tilde{D}_{F^{-1}} \\ \tilde{B}_{F^{-1}} \end{bmatrix}, \begin{bmatrix} C & \tilde{C}_{F^{-1}} \end{bmatrix}, \tilde{D}_{F^{-1}} \right) \end{aligned} \tag{24}$$

Now, the final combined model structure of both G and H is adopted as,

$$\begin{aligned} [\, G \;\; HF_k^{-1} \,] = \Bigg(&\begin{bmatrix} A & K\tilde{C}_{F^{-1}} \\ 0 & \tilde{A}_{F^{-1}} \end{bmatrix}, \\ &\begin{bmatrix} B & K \\ 0 & \tilde{B}_{F^{-1}} \end{bmatrix}, \begin{bmatrix} C & \tilde{C}_{F^{-1}} \end{bmatrix}, \begin{bmatrix} D & I \end{bmatrix} \Bigg) \end{aligned} \tag{25}$$

To solve this structured system identification problem, a grey box identification method, for example 'idgrey' in Matlab, can be used.

The overall iterative procedure can be described as follows.

(1) Use the proposed k-step-ahead subspace identification method to obtain the initial state space model (A, B, C, D, K).
(2) Obtain F_k from the noise model $H = C(qI - A)^{-1}K + I$.
(3) Apply the structured identification approach to minimize the prediction error for (25) in order to obtain new (A, B, C, D, K).
(4) Obtain a new prefilter F_k from the new noise model H.
(5) Go back to step 3. Continue until the model converges.

3. CASE STUDY

3.1 *CST Tanks in Series with A Recycle Loop*

The example chosen for illustrative purposes involves a 6 CST mixers and 1 plug flow pipe connected in series, as shown in Fig 1. In addition,

there is a recycle flow, from mixer 6 back to mixer 1. The flowrate of the secondary inlet, represented by F_u, is assumed to be the manipulated input. The concentration of the main inlet flow C_{Ad}, is treated as an unknown disturbance variable. Outputs are C_{A1} and C_{A6}. The steady state condition is $F_u = 20$, $F_d = 100$, $F_r = 200$, $C_{Ad} = 2$, and $C_u = 20$. The volume of each mixer is 1000. The dynamics of the plug flow pipe between mixer 3 and mixer 4 are represented as a pure delay of 10 time units. We assume that C_{A1} is measured and we are interested in using this measurement to inferentially control the downstream concentration C_{A6}.

First identification data are generated by performing simulations with random input variables. Both the manipulated input and the unknown disturbance variable are drawn from uniform distributions with standard deviations of 15 and 0.5 respectively and switching probability of 0.2. 50 data sets are generated for identification, each with 4000 data points. First, to test the quality of the deterministic part of the identified models, two data sets are generated with manipulated input movement only, one with a step input change and the other with random input changes. Next, to test the model-based inferential prediction and control performance, additional 1000 data points are generated with the same type of input and disturbance variations as those used to generate the 50 modeling data sets.

3.2 *Simulation Results*

The conventional subspace ID method (N4SID) and the modified subspace ID method (k-N4SID) are applied to each of the 50 data sets, which resulted in 50 pairs of state-space models. For the both identification approaches, models with 8 states are identified, and k is chosen to be 50 in applying the k-N4SID algorithm.

The resulting 50 pairs of models are first tested on the two data sets with MV movement only. After that, the identified models are tested for their final purpose, inferential prediction and control. These are done with the validation data set involving the stochastic disturbances. The control objective is to regulate the concentration of the last mixer at the steady-state value. For this, model predictive controllers are designed based on the identified models. The controllers decide the adjustments in the MV based on the inferentially predicted values of the concentration of the last mixer. For every MPC controller, the prediction horizon is chosen to be 200 time units and the control horizon is chosen to be 10 time units. Also, the input and output weighting parameters are chosen to be 10^{-7} and 1, respectively.

Table 1. Comparison of inferential prediction performances of the k-N4SID and N4SID models obtained from the 50 modeling data sets

	1-step inference			k-step inference		
	mean	min	max	mean	min	max
N4SID	0.6259	0.1732	3.8915	0.7835	0.1593	6.6758
k-N4SID	0.4188	0.1812	0.9344	0.4183	0.1457	0.9691

The benefits of the proposed modification to the subspace identification method are clearly seen in the statistical comparison involving the 50 pairs models obtained with N4SID and k-N4SID. First, N4SID resulted in more unstable models, 28 compared to 23 by the k-N4SID. Unstable models for a stable system do not necessarily lead to bad prediction and control performance as long as a stable predictor is formed. However, depending on the location of the unstable eigenvalues, extremely poor prediction and control performance can result, even though the predictor may be stable. It was observed that none of the unstable models obtained by k-N4SID resulted in bad inferential prediction and control, whereas many unstable models obtained by N4SID led to very poor inferential prediction and control results, implying the unstable modes for the N4SID models were much faster growing than those found in the k-N4SID models. Table 1 shows the better performance by the k-N4SID models over the N4SID models, in terms of both one-step inferential prediction and k-step inferential prediction. The subsequent inferential control tests also confirmed the superior quality of the models by k-N4SID over those by N4SID.

To further scrutinize the differences, the identified models were grouped in four categories according to whether both or one of the N4SID and k-N4SID methods resulted in an unstable model. For all four categories, models obtained by k-N4SID method showed better overall inferential prediction and control performance than the corresponding models by the N4SID method. This includes the cases, where N4SID gave a stable model but k-N4SID gave an unstable model. Due to space limit, only the result from the first category, for which the data sets resulted in stable models with k-N4SID but unstable models with N4SID, is shown here. The unstable nature of the models from conventional N4SID can clearly be seen from Figure 2, which shows for one of the data sets the *open-loop* predictions of the two models for a step change in the MV. Figures 3 and 4 display the corresponding differences in the inferential prediction and control performances. We can see that significant improvements in inferential prediction and control performances could be achieved by using k-N4SID instead of N4SID.

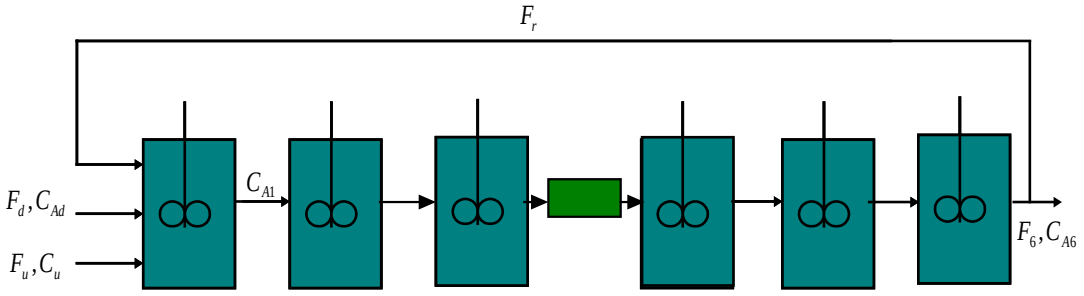


Fig. 1. The schematic for the example of 6 CST mixers in series with a recycle loop

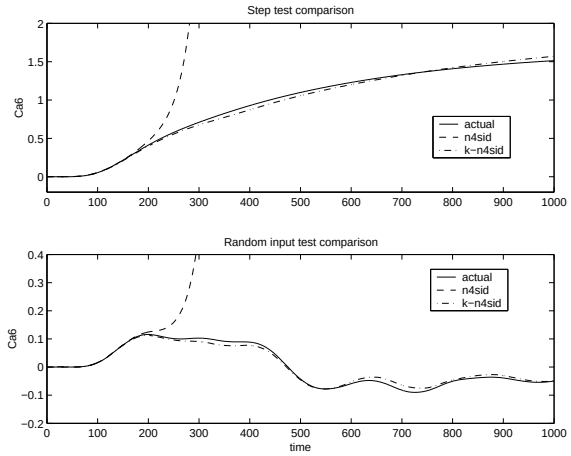


Fig. 2. Deterministic model comparison based on the MV movement data for case 1

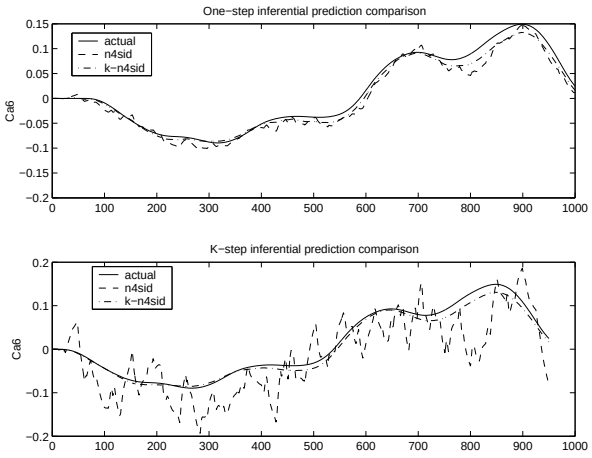


Fig. 3. Inferential prediction performance comparison for case 1

Acknowledgement: The Authors gratefully acknowledge the financial support from the National Science Foundation (CTS-#0096326).

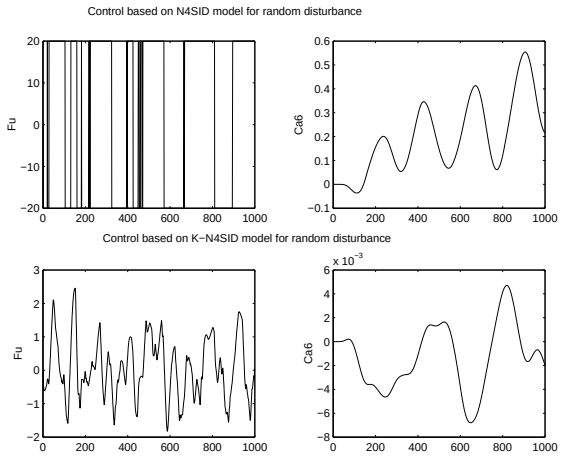


Fig. 4. Inferential control performance comparison for case 1

4. REFERENCES

[1] Ljung, L., *System Identification: Theory For the User*, Prentice-Hall, 2nd edition, 1999.

[2] Shook, D., Mohtadi, S., C., and Shah, S. L., Identification for long-range predictive control, *IEE-Proceedings-*, 138, 75-84, 1991.

[3] Huang, B., Malhotra, A., and Tamayo, E.C., Model predictive control relevant identification and validation, submitted to *Chemical Engineering and Science*, 2001.

[4] Ljung, L., Estimation focus in system identification: prefiltering, noise models, and prediction, *Proceedings of the 38th Conference on Decision and Control*, Phoenix, Arizona, USA, 2810-2815, 1999.

[5] Chaplais, F., and Alaou, K., Two time scaled parameter identification by coordination of local identifiers, *Automatica*, Vol. 32, No. 9, 1303-1309, 1996.

[6] Wahlberg, B., and Ljung, L., Design variables for bias distribution in transfer function estimation, *IEEE Trans. Automatic Control*, AC-31, 134-144, 1986.

[7] Luse, D. W., and Khalil, H., Frequency domain results for systems with slow and fast dynamics,*IEEE Trans. Automatic Control*, AC-30, 1171-1179, 1985.

[8] Van Zee, G. A., and Bosgra, O. H., Validation of prediction error identification results by multivariable control implementation, in *Proc. 6th IFAC Symp. Ident. Syst. Parameter Estimation*, Washington, DC, 560-565, 1982.

[9] Van Overschee, P. and DeMoor, B., *Subspace Identification for Linear Systems: Theory, Implementation, Applications.*, Kluwer Academic, Dordrecht, The Netherlands, 1996.

[10] Åström, K. J., Maximum likelihood and prediction error methods, *Automatica*, Vol 16, 551-574, 1980.

A SUBSPACE APPROACH TO MIMO CONTROL PERFORMANCE MONITORING AND DIAGNOSIS

S Joe Qin [*,1] **Christopher A. McNabb** [**]

[*] *Department of Chemical Engineering, The University of Texas at Austin, Austin, TX 78712 USA*
[**] *Boise Paper Solutions, PO Box 50, Boise, ID 83728*

Abstract: In this paper we begin with a state space model of a generally non-square process and derive the minimum achievable variance in a state feedback form. We propose a simple control performance calculation which uses orthogonal projection of filtered output data onto past closed-loop data. Finally, we propose a control performance monitoring technique based on the output covariance and diagnose the cause of suboptimal control performance using generalized eigenvector analysis. The proposed methods are demonstrated on an industrial wood waste burning power boiler.

Keywords: MIMO control performance monitoring; generalized eigenanalysis; covariance-based monitoring

1. INTRODUCTION

With the initial success of minimum variance single-loop performance assessment (Harris, 1989; Desborough and Harris, 1992; Qin, 1998; Harris *et al.*, 1999; Kozub, 1996; Harris and Seppala, 2002) and industrial case studies (Thornhill *et al.*, 1999; Miller *et al.*, 1998; Harris *et al.*, 1996*b*; Perrier and Roche, 1992; Weinstein, 1992; Desborough and Miller, 2002), research interest has shifted to the assessment of MIMO control systems using the minimum variance benchmark (Harris *et al.*, 1996*a*; Huang *et al.*, 1997; Huang, 1997; Shah *et al.*, 2002) and a covariance-based benchmark (McNabb and Qin, 2003)

Currently the MIMO performance monitoring benchmark has been a straightforward extension of the SISO variance ratio, which looks at only the trace of the covariance matrix. However, trace based monitoring index is insufficient for assessing the multivariate covariance of the control performance. Another drawback of the existing literature is that research emphasis has been placed on control performance assessment and little has been done regarding diagnosis.

The control performance monitoring tehcniques typically calculate a benchmark performance from closed-loop operation data based on some minimal knowledge, such as the time delay information. Due to interaction, the performance suboptimality in each variable is not independent from each other. Therefore, the suboptimality of a MIMO control system necessarily occupies a subspace instead of the entire output space. In this paper we propose a subspace approach to extract the major directions of suboptimality (MDS) and measure the variance inflation in each of the directions. To deal with the MIMO control performance diagnosis, we propose to use generalized eigenvectors to diagnose the directions of suboptimality.

[1] Author for correspondence. Email: qin@che.utexas.edu Tel: (512) 471-4417 Fax: (512) 471-7060

This paper is organized as follows. The extended state space model is given in Section 2. The minimum variance control solution in state space is shown in Section 3. Section 4 introduces the view of MVC as an optimal subspace and Section 5 proposes covariance-based monitoring and demonstrates the use of generalized eigenvector diagnosis techniques in this new framework with an industrial example. The paper ends with a few concluding remarks.

2. EXTENDED STATE SPACE PROCESS MODEL

We assume the open loop process is described by the following state space model in innovation form:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Ke(k) \\ y(k) &= Cx(k) + Du(k) + e(k) \end{aligned} \tag{1}$$

where $x \in \Re^n$, $u \in \Re^m$, $y \in \Re^p$, $e \in \Re^p$ are the state, input, output and innovation vectors. A, B, C, D, and K are matrices with appropriate dimensions. Denoting

$$y_{d+1}(k) = \begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+d) \end{bmatrix}, u_{d+1}(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+d) \end{bmatrix}$$

$$e_{d+1}(k) = \begin{bmatrix} e(k) \\ e(k+1) \\ \vdots \\ e(k+d) \end{bmatrix}$$

$$\Gamma_{d+1} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{d-1} \\ CA^d \end{bmatrix}$$

$$H_{d+1} = \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & D & 0 \\ CA^{d-1}B & CA^{d-2}B & \cdots & CB & D \end{bmatrix}$$

$$G_{d+1} = \begin{bmatrix} I & 0 & 0\cdots & \cdots & 0 \\ CK & I & 0 & \cdots & 0 \\ CAK & CK & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & I & 0 \\ CA^{d-1}K & CA^{d-2}K & \cdots & CK & I \end{bmatrix}$$

where d is the time delay order of this multivariate system. The notation $y_j(k)$ and $e_j(k)$ will be used throughout this paper to represent j-element vectors of $y(k)$ and $e(k)$ extending from time k to $k+j-1$.

We have the following extended equation:

$$\begin{aligned} y_{d+1}(k) = &\Gamma_{d+1}x(k) + H_{d+1}u_{d+1}(k) \\ &+G_{d+1}e_{d+1}(k) \end{aligned} \tag{2}$$

3. STATE SPACE MINIMUM VARIANCE CONTROL

Because of the time delay, not all elements of $y_{d+1}(k)$ are affected by $u(k)$. McNabb (McNabb, 2002) and McNabb and Qin (McNabb and Qin, 2003) present a new algorithm for deriving a multivariate time delay (MTD) matrix $L \in \Re^{(d+1)p \times d}$ for use with the extended state space model (2).

Denote

$$g_{d+1} \triangleq \begin{bmatrix} I \\ CK \\ CAK \\ \vdots \\ CA^{d-1}K \end{bmatrix}, \quad h_{d+1} \triangleq \begin{bmatrix} D \\ CB \\ CAB \\ \vdots \\ CA^{d-1}B \end{bmatrix}$$

$$G_{d+1} = \left[g_{d+1} \left| \begin{matrix} 0 \\ \hline G_d \end{matrix} \right. \right], \quad H_{d+1} = \left[h_{d+1} \left| \begin{matrix} 0 \\ \hline H_d \end{matrix} \right. \right],$$

$$e_d(k+1) = \begin{bmatrix} e(k+1) \\ \vdots \\ e(k+d) \end{bmatrix}$$

To extract the multivariate time delay, the algorithm proposed by McNabb (McNabb, 2002) uses the first few Markov parameters or the {A, B, C, D} matrices, which shifts the output $y(k)$ forward successively and forms a new output by a linear combinations of the shifted outputs,

$$\begin{aligned} y^{(j)}(k) &= L^{(j)} y_{j+1}(k) \\ &= C^{(j)}x(k) + D^{(j)}u(k) \end{aligned}$$

The matrix $L^{(j)} \in R^{P \times (j+1)P}$ is chosen such that $D^{(j)}$ keeps the maximum possible rank. The algorithm will terminate when $D^{(j)}$ reaches full column rank. At the end of the iteration, set

$$\begin{aligned} d &= j \\ L &= L^{(j)} = L^{(d)} \\ \Lambda &= D^{(j)} = D^{(d)} \\ \tilde{y}(k+d) &= y^{(d)}(k) \end{aligned}$$

The output $\tilde{y}(k+d)$ is known as the MTD-filtered output which has the following expression,

$$\begin{aligned}\tilde{y}(k+d) &= Ly_{d+1}(k) \\ &= L\Gamma_{d+1}x(k) + \Lambda u(k) + Lg_{d+1}e(k) \\ &\quad + L\begin{bmatrix} 0 \\ G_d \end{bmatrix} e_d(k+1) \end{aligned} \tag{3}$$

The minimum variance control of system (1) is achieved by

$$u(k) = -\Lambda^{+}L(\Gamma_{d+1}x(k) + g_{d+1}e(k)) \tag{4}$$

where Λ^{+} is the Moore-Penrose pseudo-inverse. The feedback invariant term or the output under minimum variance control is

$$\tilde{y}_{mv}(k+d) = L\begin{bmatrix} 0 \\ G_d \end{bmatrix} e_d(k+1)$$

The filtered output shown in Equation (3) can be interpreted as the combination of two independent terms; an optimal d step ahead prediction of $\tilde{y}(k+d)$ and the associated prediction error

$$\tilde{y}(k+d) = \tilde{y}(k+d|k) + \tilde{y}_{mv}(k+d) \tag{5}$$

where

$$\tilde{y}(k+d|k) = L\Gamma_{d+1}x(k) + \Lambda u(k) + Lg_{d+1}e(k) \tag{6}$$

represents the optimal d step ahead prediction of $\tilde{y}(k+d)$.

McNabb and Qin (McNabb and Qin, 2003) show further that L corresponds to a unitary interactor and

$$E\left(\tilde{y}^T(k)\tilde{y}(k)\right) = E\left(y^T(k)y(k)\right)$$

In other words, the sum of the variance of each original output variable is minimized by the MVC law in Eq. 4.

4. CALCULATION OF MVC VARIANCE BY SUBSPACE PROJECTION

The optimal prediction $\tilde{y}(k+d|k)$ under a time-invariant controller is related to (McNabb, 2002),

$$\tilde{y}(k+d|k) = \Theta_r y_r(k-r+1) \tag{7}$$

where $y_r(k-r+1) = \left[y^T(k-r+1), \cdots, y^T(k)\right]^T$ and r is sufficiently large. As a consequence, Equation (5) can be rewritten as

$$\tilde{y}(k+d) = \Theta_r y_r(k-r+1) + \tilde{y}_{mv}(k+d) \tag{8}$$

Again $y_r(k-r+1)$ depends on data before time k and $\tilde{y}_{mv}(k+d)$ depends only on innovations from time $(k+1)$ through $(k+d)$, which are independent of $y_r(k-r+1)$. Therefore, $E\{\tilde{y}_{mv}(k+d)\cdot y_r^T(k-r+1)\} = 0$. Formulating three data matrices with column dimension N

$$\begin{aligned}\tilde{Y}_N &= \begin{bmatrix}\tilde{y}(k+d) & \tilde{y}(k+d+1) \\ \cdots & \tilde{y}(k+d+N-1)\end{bmatrix} \\ \tilde{Y}_{mv,N} &= \begin{bmatrix}\tilde{y}_{mv}(k+d) & \tilde{y}_{mv}(k+d+1) \\ \cdots & \tilde{y}_{mv}(k+d+N-1)\end{bmatrix} \\ Z_{r,N} &= \begin{bmatrix}y_r(k-r+1) & y_r(k-r+2) \\ \cdots & y_r(k-r+N)\end{bmatrix}\end{aligned}$$

we have $\frac{1}{N}\tilde{Y}_{mv,N}Z_{r,N}^T \to E\{\tilde{y}_{mv}(k+d)y_r^T(k-r+1)\} = 0$ as $N \to \infty$. Therefore, defining $\Pi_Z^{\perp} = I - Z_{r,N}^T\left(Z_{r,N}Z_{r,N}^T\right)^{-1}Z_{r,N}$, we have

$$\tilde{Y}_{mv,N} = \tilde{Y}_N\Pi_Z^{\perp}$$

The MVC covariance is

$$cov(\tilde{y}_{mv}(k)) = \frac{1}{N-1}\tilde{Y}_{mv,N}\tilde{Y}_{mv,N}^T \quad \text{as } N \to \infty \tag{9}$$

and the associated multivariate control performance index is

$$\eta = \frac{tr\left\{cov(\tilde{y}_{mv}(k))\right\}}{tr\left\{cov(\tilde{y}(k))\right\}} = \frac{tr\left\{\tilde{Y}_{mv,N}\tilde{Y}_{mv,N}^T\right\}}{tr\left\{\tilde{Y}_N\tilde{Y}_N^T\right\}}$$

From the above derivation we demonstrate that:

(1) The output of the process under minimum variance control can be calculated by a single row projection of the MTD filtered output data onto the row space of the normal closed-loop output data, and
(2) The minimum variance output occupies an *optimal* subspace of the general closed-loop output.

The variance index has a value between 0 and 1, with 1 corresponding to the minimum variance. The limitation of η, however, is that it only considers the trace of the covariance matrix, ignoring the off-diagonal terms of the covariance.

5. PERFORMANCE MONITORING BASED ON OUTPUT COVARIANCE

Most of the MIMO performance indices are based on the sum of variances of each output, i.e., the trace of the covariance matrix. This type of index is adequate when all variables are fairly independent. In practice, however, the output variables are rarely independent, especially in the case of ill-conditioned plants and highly interacting plants. In these cases, it is more appropriate to use the covariance of the output to monitor controller

performance. The benchmark covariance can be the minimum variance benchmark, but it can be any other benchmarks.

5.1 Covariance-based Indices and Suboptimality Directions

To find a direction in $\tilde{y}(k)$ along which the worst suboptimality occurs, we find the direction p with $\|p\| = 1$ and project $\tilde{y}(k)$ and $\tilde{y}_{mv}(k)$ to this direction:

$$\begin{aligned} \Pi_p\tilde{y}(k) &= p^T\tilde{y}(k)/p^Tp &= p^T\tilde{y}(k) \\ \Pi_p\tilde{y}_{mv}(k) &= p^T\tilde{y}_{mv}(k)/p^Tp &= p^T\tilde{y}_{mv}(k) \end{aligned}$$

The variance of the projections are, respectively,

$$\begin{aligned} var(\Pi_p\tilde{y}(k) &= p^T cov(\tilde{y}(k))p \\ var(\Pi_p\tilde{y}_{mv}(k) &= p^T cov(\tilde{y}_{mv}(k))p \end{aligned}$$

The direction p along which the largest variance ratio occurs is

$$p = arg\ max \frac{p^T cov(\tilde{y}(k))p}{p^T cov(\tilde{y}_{mv}(k))p} \tag{10}$$

The direction of p after maximization shows the direction with the most potential to improve the performance. The solution to this problem is a generalized eigenvector problem,

$$cov(\tilde{y}(k))p_i = \mu_i cov(\tilde{y}_{mv}(k))p_i$$

where p_i is the generalized eigenvector corresponding to the i^{th} largest generalized eigenvalue μ_i. The "volume" of the suboptimality or variance inflation due to poor control performance is:

$$\prod_{i=1}^{l} \mu_i$$

where l is the number of selected directions. The volume-based performance can be defined as

$$I_v(l) = \left(\prod_{i=1}^{l} \mu_i \right)^{-1}$$

McNabb and Qin (McNabb and Qin, 2003) show that for all possible projections Π,

$$cov\left(\Pi\tilde{y}_{mv}(k)\right) \leq cov\left(\Pi\tilde{y}(k)\right)$$

Therefore, $\mu_i \geq 1$ and I_v is between zero and one. When $\tilde{y}(k)$ achieves the minimum variance performance, I_v approaches one. On the other hand, I_v close to zero indicates a very poor performance.

The volume based performance index can be very small due to the multiplication effect of several small numbers. To normalize this effect, we define a radius-based performance index as follows:

$$I_r(l) = (I_v(l))^{1/l} = \left(\prod_{i=1}^{l} \mu_i \right)^{-1/l}$$

This index also ranges between zero and one. It provides a geometric average of the poor performance in all l directions. Note that $I_v(l)$ and $I_r(l)$ consider the covariance matrices of $\tilde{y}(k)$ and $\tilde{y}_{mv}(k)$, whereas the η index focuses on variance only.

After a significant suboptimality is detected by using $I_v(l)$ or $I_r(l)$, the major directions of suboptimality are already calculated as p_i. These directions can then be used to locate the main sources of suboptimality.

5.2 Industrial Example

This example uses industrial data from a wood waste (*hog fuel*) burning power boiler. Five second samples of process variables (PV) with their corresponding setpoints (SP) and controller outputs (OP) were collected from the DCS over an eleven day period. The process and instrumentation diagram of the boiler process is shown in Figure 1. We selected five PV's with associated SP's as shown in Table 1. Full open loop testing was not possible on the power boiler. We therefore assume that each PV had a single time delay associated with the full set of manipulated variables, corresponding to a diagonal interactor. The individual time delays for each of these loops (in units of sample periods) were 1, 34, 1, 2 and 1, respectively. All analysis was performed on data scaled to zero mean and unit variance.

Table 1. Power boiler tags used in analysis

Variable #	Tag	Description
1	FC1	Total air flow
2	PC1	Boiler master (900# header pressure)
3	PC2	Forced draft fan pressure
4	PC3	Furnace pressure
5	PC4	Overfire air pressure

Time series plots of $(PV - SP)$ for each of the five loops are shown in Figure 2 and are divided into 50 sequential 250 minute periods.

To apply the covariance based monitoring to this problem we first select the significant number of principal components of the five process variables. We choose the number of PC's to be four and calculate the volumetric performance index (I_v) and the radius-based performance index (I_r) as shown in Figure 3. Both indices show a clear drop from Period I to Period II. For Period I the I_r index is around 0.6, which means on average each

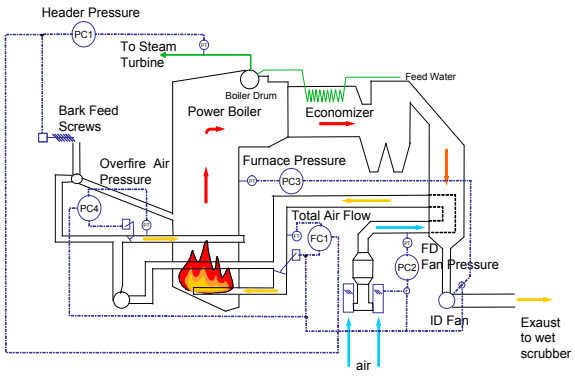


Fig. 1. Power boiler schematic

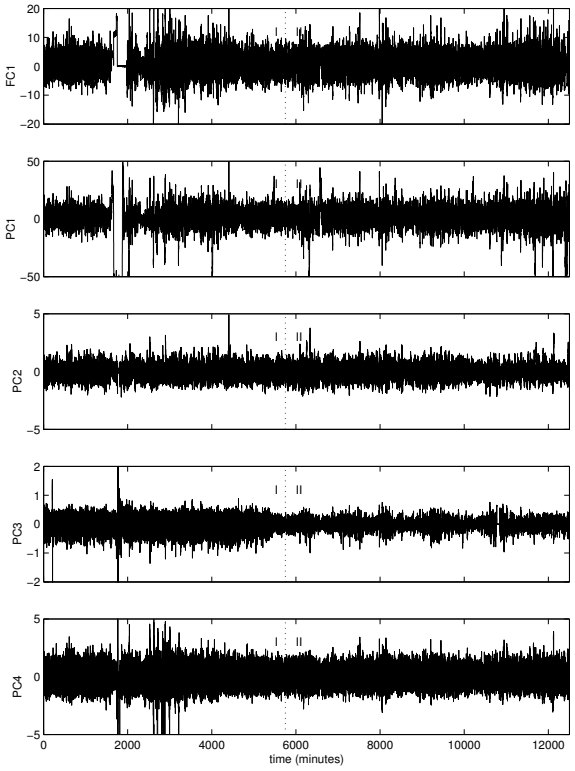


Fig. 2. Time series of setpoint error ($PV - SP$) for five power boiler loops

variable is about 60% of its optimal performance. For Period II the average performance drops to about 40% of its optimal performance.

In reality, however, it is usually the case that some loops are worse performers than others. To identify the directions of suboptimality and the contributing variables, two 250 minute blocks were chosen for more detailed analysis. The period extending from 3500 minutes to 3750 minutes (the 15th analysis block) is used to represent the behavior before the shift in control performance and the period extending from 10000 minutes to 10250 minutes (the 41st analysis block) is used to represent the system behavior following the shift in control performance. The generalized eigenvector analysis was performed separately on these two periods as shown in Equation (10). The upper and

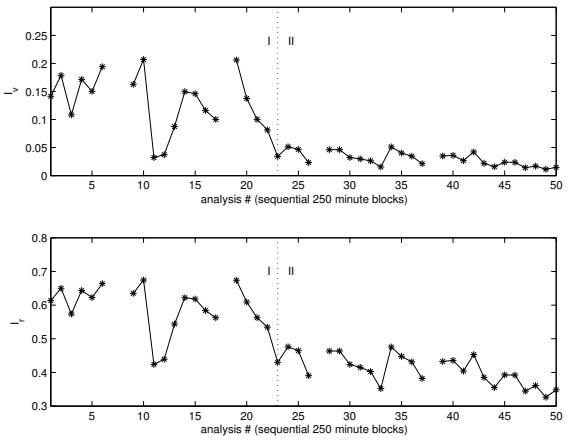


Fig. 3. Trends of the I_v index (top plot) and the I_r (bottom plot) index for 4 retained principal components ($l = 4$)

lower plots on the left of Figure 4 show the eigenvalues for the two time periods (labeled as dataset #1 and dataset #2). The middle plots show the first eigenvectors and the right plots show the second eigenvectors for both time periods. It is clear that for both time periods, the major suboptimality lies in two directions, although there are five controlled variables. The first direction of suboptimality is dominant in both time periods, with dataset #1 about 70-fold and dataset #2 about 110-fold for potential improvement. Since the suboptimality is adequately captured in two directions, the covariance-based MIMO performance monitoring indicates that the suboptimality in the five controlled variables is highly correlated. It is possible that by improving the performance of one loop, the other loops are improved due to correlation or interaction. For both time periods the directions of suboptimality point to variable 1 (FC1) for the most potential to improve and variable 2 (PC1) for the least potential to improve. It is likely that by improving the performance of FC1

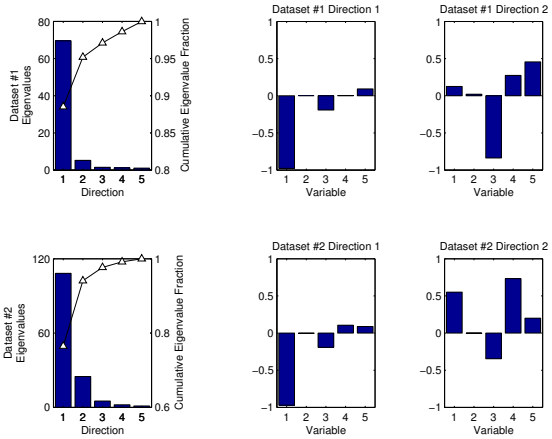


Fig. 4. Eigenvector decomposition of $cov(\tilde{y}_{mv}(k))^{-1}cov(\tilde{y}(k))$ for dataset # 1 (top 3 plots) and dataset # 2 (bottom 3 plots).

other loops will be improved due to interaction. By examining the process diagram in Figure 1, it is confirmed that this is likely because FC1 is the total air flow which directly affects all other loops except PC1, which is the steam header pressure. A significant shift in the second direction occurs between datasets #1 and #2. In dataset #1 PC2 dominates the second eigenvector but in dataset #2 FC1 and PC3 dominate.

6. CONCLUSIONS

The main contributions of this paper are the use of covariance based monitoring and the application of a generalized eigenvector based technique for identifying the major directions of suboptimality of MIMO feedback control systems. This framework provides a systematic performance diagnosis method as well as covariance-based performance assessment indices. Future work will focus on the impact of sensor and actuator faults on control performance.

7. ACKNOWLEDGMENTS

Financial support for this work from the National Science Foundation under CTS-9814340, Texas Higher Education Coordinating Board, and Weyerhaeuser Company through sponsorship of the Texas Modeling and Control Consortium is gratefully acknowledged.

REFERENCES

Desborough, L. and T. J. Harris (1992). Performance assessment measures for univariate feedback control. *Can. J. Chem. Eng.* **70**, 1186.

Desborough, Lane and Randy Miller (2002). Increasing customer value of industrial control performance monitoring – honeywell's experience. In: *Chemical Process Control - CPC VI*. CACHE. Tuscon, Arizona. pp. 169–189.

Harris, T., F. Boudreau and J.F. Macgregor (1996*a*). Performance assessment of multivariable feedback controllers. *Automatica* **32**(11), 1505–1518.

Harris, T. J. (1989). Assessment of control loop performance. *Can. J. Chem. Eng.* **67**(10), 856–861.

Harris, T. J. and C. T. Seppala (2002). Recent developments in controller performance monitoring and assessment techniques. In: *Chemical Process Control - CPC VI*. CACHE. Tuscon, Arizona. pp. 208–222.

Harris, T.J., C.T. Seppala and L.D. Desborough (1999). A review of performance monitoring and assessment techniques for univariate and multivariate control systems. *J. Proc. Cont.* **9**, 1–17.

Harris, T.J., C.T. Seppala, P.J. Jofriet and B.W. Surgenor (1996*b*). Plant-wide feedback control performance assessment using an expert system framework. *Control Engineering Practice* **4**(9), 1297–1303.

Huang, B. (1997). Multivariate Statistical Methods For Control Loop Performance Assessment. PhD thesis. University of Alberta.

Huang, B., S.L. Shah and K.Y. Kwok (1997). Good, bad or optimal? performance assessment of MIMO processes. *Automatica* **33**(6), 1175–1183.

Kozub, D.J. (1996). Controller performance monitoring and diagnosis: experiences and challenges. In: *Fifth Int. Conf. on Chemical Process Control* (J.C. Kantor, C.E. Garcia and B.C. Carnahan, Eds.). AIChE and CACHE. Tahoe, CA. pp. 83–96.

McNabb, C. A. and S. J. Qin (2003). Projection based MIMO control performance monitoring – I. Covariance monitoring in state space. *J. Proc. Cont.* Accepted.

McNabb, C.A. (2002). MIMO Control Performance Monitoring Based on Subspace Projections. PhD thesis. The University of Texas at Austin.

Miller, R., L. Desborough and C. Timmons (1998). Citgo's experience with controller performance assessment. In: *NPRA 1998 Computer Conference*. San Antonio, Texas.

Perrier, M. and A. Roche (1992). Towards mill-wide evaluation of control loop performance. In: *Control Systems '92*. Whistler, British Columbia.

Qin, S.J. (1998). Control performance monitoring – a review and assessment. *Comput. Chem. Eng.* **23**, 178–186.

Shah, S.L., R. Patwardhan and B. Huang (2002). Multivariate controller performance assessment: methods, applications and challenges. In: *Chemical Process Control - CPC VI*. CACHE. Tuscon, Arizona. pp. 190–207.

Thornhill, N.F., M. Oettinger and P. Fedenczuk (1999). Refinery-wide control loop performance assessment. *J. Proc. Cont.* **9**, 109–124.

Weinstein, B. (1992). A sequential approach to the evaluation and optimization of control system performance. In: *1992 ACC*. Chicago. pp. 2354–2358.

MULTIVARIATE CONTROLLER PERFORMANCE ASSESSMENT WITHOUT INTERACTOR MATRIX — A SUBSPACE APPROACH

Ramesh Kadali and Biao Huang [1]

Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB, Canada T6G 2G6

Abstract: Several methods for multivariate control performance assessment (MPA) with or without using the interactor matrix have been proposed in the literature. They are all equivalent, one way or other, by certain transformations. In this paper a subspace framework for MPA is proposed for the estimation of MVC-benchmark variance for feedback multivariate systems. The merit of the new approach is that we start straight from data, and a performance index is calculated directly from subspace matrices without relying on a parametric dynamic model. In addition, a proof that the proposed solution is exactly the same as that of the conventional approaches is provided.

1. INTRODUCTION

Periodic performance assessment of the controllers is important for maintaining normal process operation and to sustain the performance of controllers achieved when the controllers are commissioned. Minimum variance control is theoretically the best possible control (Astrom and Wittenmark, 1984). Controller performance assessment using MVC-benchmark involves comparing the current process output variance with the output benchmark variance if a minimum variance controller were implemented on the process. Although the intention of many industrial controllers is not minimum variance control, MVC-benchmark is nevertheless used as a first step in the controller performance assessment (Harris, 1989). Calculation of the MVC-benchmark variance for univariate systems from routine closed loop data requires *a priori* knowledge of only the process time delay (Harris, 1989; Huang and Shah, 1999). Calculation of the MVC-benchmark variance for multivariate systems involves calculation of the interactor matrix (Huang and Shah, 1999) for the system from the first few process Markov parameters. Furthermore, the concept of the interactor is not well known in practice. Hence, estimation of the MVC-benchmark without the interactor matrix has been an active area of research.

(Ko and Edgar, 2001) proposed a method for the estimation of the multivariate MVC-benchmark using closed loop data, which does not require the intermediate interactor matrix calculation. It is shown that their result is equivalent to the result of (Huang and Shah, 1999). Recent progress in the subspace approach to closed loop identification (Kadali and Huang, 2002) inspires an alternative approach for the estimation of multivariate MVC-benchmark. In the current paper we will show the estimation of the multivariate MVC-benchmark with *neither* the interactor matrix calculation *nor* the Markov parameters. The only *a priori* knowledge required is the deterministic subspace matrix directly calculated from data. The important difference between the "calculation of the subspace

[1] The author to whom all correspondence should be addressed. Phone: (780)492-9016, FAX: (780)492-2881. E-mail:bhuang@ualberta.ca

matrix" and subspace identification is that the former does not extract an explicit "model" and is also known as model-free approach in the literature. This will further simplify the procedure for the calculation of the multivariate performance index. No concepts such as interactor matrix, Markov parameter, multivariate transfer function matrix, state space model etc. are needed to apply this technique and this will make the multivariate controller performance assessment technique more applicable in practice. (McNabb and Qin, 2001) have also proposed another subspace approach to multivariate performance monitoring by projecting delay-matrix filtered output data onto past data.

In the subspace method, certain subspace matrices are identified as a first step in the subspace identification methods. The minimum variance controller can be designed directly using these intermediate subspace matrices, without a parametric model such as the state space model or transfer function model. The closed loop subspace identification method proposed in (Kadali and Huang, 2002) allows a convenient identification of subspace matrices from the closed loop data with external excitations. The MVC-benchmark variance can be calculated with the knowledge of only the deterministic subspace matrix and eliminates the intermediate step of estimating the unitary interactor matrix or extracting the Markov parameters.

We do not claim that the subspace approach as proposed in this paper requires less a priori knowledge than other methods. In fact, the equivalent information of the interactor matrix or Markov parameters is implicitly buried in the subspace matrices. However, avoiding direct use of the interactor matrix and/or Markov parameter matrices does have an advantage of easier acceptance by practitioners and reduces unnecessary intermediate modeling step. Another merit of this paper lies in the direct data based approach, i.e. from process experiment data, a multivariate performance index is directly calculated. Comparing with the conventional methods such as that proposed by (Huang and Shah, 1999), our method is different in the sense of subspace approach versus conventional transfer function approach. The method proposed by (Ko and Edgar, 2001), even though without using the interactor matrix, is nevertheless following the transfer function matrix approach and is an extension of (Harris *et al.*, 1996; Huang and Shah, 1999). Our approach, which may be considered an extension to (Ko and Edgar, 2001), adopts the subspace framework, and further avoids the use of the transfer function matrix and Markov parameters.

2. SUBSPACE MATRICES DESCRIPTION

Consider the following innovations state space representation of a linear time-invariant system with l-inputs (u_k), m-outputs (y_k) and n-states (x_k) as:

$$x_{k+1} = Ax_k + Bu_k + Ke_k \tag{1}$$

$$y_k = Cx_k + e_k \tag{2}$$

where the state space system matrices A, B, C and K^f are $(n \times n)$, $(n \times l)$, $(m \times n)$ and $(n \times m)$ matrices respectively. K^f is the Kalman filter gain and e_k is an unknown innovation sequence.

The matrix input-output equations used in subspace identification (Overschee and Moor, 1994; Overschee and Moor, 1995; Overschee and Moor, 1996) expressed using certain subspace matrices L_w, L_u and L_e (Overschee and Moor, 1996) as

$$y_f = L_w w_p + L_u u_f + L_e e_f \tag{3}$$

where

$$y_f = \begin{bmatrix} y_{t+1} \\ ... \\ y_{t+N} \end{bmatrix}; y_p = \begin{bmatrix} y_{t-N+1} \\ ... \\ y_t \end{bmatrix}; e_f = \begin{bmatrix} e_{t+1} \\ ... \\ e_{t+N} \end{bmatrix};$$

$$u_f = \begin{bmatrix} u_{t+1} \\ ... \\ u_{t+N} \end{bmatrix}; u_p = \begin{bmatrix} u_{t-N+1} \\ ... \\ u_t \end{bmatrix}; w_p = \begin{bmatrix} y_p \\ u_p \end{bmatrix}$$

The subspace matrices are estimated as an intermediate step by data projections (Overschee and Moor, 1996). L_u and L_e are dynamic matrices containing the estimated Markov parameters corresponding to the process and noise respectively.

Recent results in subspace closed-loop identification (Kadali and Huang, 2002) allow the direct estimation of two of the subspace matrices, L_u and L_e, from the closed loop data with set point excitation. Note that although the deterministic subspace matrix and closed loop noise matrix contain process Markov parameters and noise Markov parameters respectively, the two matrices are directly calculated from closed-loop data by a projection method and one never needs to know what are inside these two matrices in order to apply our algorithms. The only reason to mention Markov parameters here and in the sequel is to analytically compare our results with conventional results available in the literature.

3. DESIGN OF MINIMUM VARIANCE CONTROL USING SUBSPACE MATRICES

The minimum variance controller (MVC) is designed to minimize the following quadratic cost function J over the horizon N, as $N \longrightarrow \infty$:

$$J = \frac{1}{N} E\{ \sum_{k=1}^{N} [(r_{t+k} - y_{t+k})^T \ (r_{t+k} - y_{t+k})] \} \quad (4)$$

$$= \frac{1}{N} \sum_{k=1}^{N} [(r_{t+k} - \hat{y}_{t+k})^T \ (r_{t+k} - \hat{y}_{t+k})] \quad (5)$$

where E is the expectancy operator, r_t is the reference for output trajectory. $\hat{y}_{t+k}$ is the k-step ahead predicted output given the past inputs and outputs upto time t.

Using equation (3), the optimal predictor equation can be written as $\hat{y}_f = L_w w_p + L_u u_f$. The notation in the cost function can be simplified for regulatory control, by letting $r_{t+k} = 0$ as:

$$J = \min_{u_f^2} \ \frac{1}{N} [\hat{y}_f^T \ \hat{y}_f]$$

$$= \min_{u_f^2} \frac{1}{N} [(L_w w_p + L_u u_f)^T (L_w w_p + L_u u_f)] \quad (6)$$

To obtain the minimum variance control law, we differentiate J with respect to u_f and set it to zero to obtain the control law as:

$$u_f = -L_u^{\dagger}(L_w w_p) \quad (7)$$

where, $\dagger$ represents pseudo-inverse. The above control law is the minimum variance control law as the number of block-rows in the subspace matrices L_w and L_u tend to infinity.

4. ESTIMATION OF THE MULTIVARIATE MVC-BENCHMARK

From the very first block-element of Y_f in equation (3) we can write

$$y_{t+1} = L_{y_p} \begin{bmatrix} y_{t-N+1} \\ ... \\ y_t \end{bmatrix} + L_{u_p} \begin{bmatrix} u_{t-N+1} \\ ... \\ u_t \end{bmatrix} + L_0 e_{t+1} \quad (8)$$

where $L_{y_p} = L_w(1:m, 1:mN)$ and $L_{u_p} = L_w(1:m, mN+1 : (l+m)N)$. Equation (8) can be transformed for an equivalent expression of y_{t+1} in terms of the past inputs and the past noise as

$$y_{t+1} = \begin{bmatrix} G_1 \ ... \ G_N \end{bmatrix} \begin{bmatrix} u_t \\ ... \\ u_{t-N+1} \end{bmatrix} + \begin{bmatrix} L_1 \ ... \ L_N \end{bmatrix} \begin{bmatrix} e_t \\ ... \\ e_{t-N+1} \end{bmatrix} + L_0 e_{t+1} \quad (9)$$

where G_i and L_i are the i-th impulse response coefficients (Markov parameters for multivariate systems) of the process and noise models respectively. In other words, we can express the past (state) contribution term, $L_w w_p$, as

$$L_w w_p = \begin{bmatrix} G_1 & ... & G_{N-1} & G_N \\ G_2 & ... & G_N & 0 \\ ... & ... & ... & ... \\ G_N & 0 & ... & 0 \end{bmatrix} \begin{bmatrix} u_t \\ ... \\ u_{t-N+1} \end{bmatrix} + \begin{bmatrix} L_1 & ... & L_{N-1} & L_N \\ L_2 & ... & L_N & 0 \\ ... & ... & ... & ... \\ L_N & 0 & ... & 0 \end{bmatrix} \begin{bmatrix} e_t \\ ... \\ e_{t-N+1} \end{bmatrix} \quad (10)$$

However, the controller output, u_{t+1} is calculated using all the data available at time '$t+1$', i.e., $\{u_t, y_{t+1}, u_{t-1}, y_t, ...\}$. Hence the original subspace predictor expression in equation (3) and the subspace based minimum variance control law in equation (7) have to be modified to obtain the closed loop expressions for u_f and y_f. First, define

$$L_G = \begin{bmatrix} G_1 & G_2 & ... & G_{N-1} & G_N \\ G_2 & G_3 & ... & G_N & 0 \\ ... & ... & ... & ... & \\ G_N & 0 & 0 & ... & 0 \end{bmatrix}; \tilde{u}_p = \begin{bmatrix} u_t \\ u_{t-1} \\ ... \\ u_{t-N+1} \end{bmatrix}$$

$$L_H = \begin{bmatrix} L_0 & L_1 & ... & L_{N-1} & L_N \\ L_1 & L_2 & ... & L_N & 0 \\ ... & ... & ... & ... & ... \\ L_{N-1} & 0 & 0 & ... & 0 \end{bmatrix}; \tilde{e}_p = \begin{bmatrix} e_{t+1} \\ e_t \\ ... \\ e_{t-N+1} \end{bmatrix}$$

$$\tilde{L}_e = \begin{bmatrix} 0 & 0 & ... & 0 \\ L_0 & 0 & ... & 0 \\ ... & ... & ... & ... \\ L_{N-2} & L_{N-3} & ... & 0 \end{bmatrix}; \tilde{e}_f = \begin{bmatrix} e_{t+2} \\ e_{t+3} \\ ... \\ e_{t+N+1} \end{bmatrix}$$

Since L_G and L_H contain the process and noise model Markov parameters, they can be formed from the subspace matrices L_u and L_e respectively. Therefore the equation based on the first column of Y_f in equation (3) can be alternatively written as

$$y_f = L_G \tilde{u}_p + L_H \tilde{e}_p + L_u u_f + \tilde{L}_e \tilde{e}_f \quad (11)$$

Substituting equation (10) in equation (7), we can write

$$u_f = -L_u^{\dagger} \ \{L_w w_p\} = -L_u^{\dagger} \{L_G \tilde{u}_p + L_H \tilde{e}_p\} \quad (12)$$

The closed loop expression for y_f can be written as

$$y_f = (I - L_u L_u^{\dagger}) \ (L_G \tilde{u}_p + L_H \tilde{e}_p) + \tilde{L}_e \tilde{e}_f \quad (13)$$

Now that we have derived closed-loop expressions for both u and y, the next step is to calculate their variance expressions which are actually the H_2 norm of the closed-loop expressions weighted

by the variance of e. A simple method to derive the variance expression is given below.

Let a disturbance enter the process at time $= t+1$, i.e., $u_t = u_{t-1} = ... = u_{t-N+1} = 0$; $e_t = e_{t-1} = ... = e_{t-N+1} = 0$; and $e_{t+2} = e_{t+3} = ... = e_{t+N} = 0$. Then the cumulative effect of the noise e_{t+1} on the process output variance can be obtained from equation (13), which simplifies to

$$y_f = (I - L_u L_u^{\dagger}) L_h \; e_{t+1} \;\; = \;\; \begin{bmatrix} \psi_0 \\ \psi_1 \\ ... \end{bmatrix} e_{t+1} \quad (14)$$

where $L_h = \begin{bmatrix} L_0 \\ ... \\ L_{N-1} \end{bmatrix}$, the vector of noise model Markov parameters, and ψ_i represents the Markov parameter of i-th lag of the closed loop noise model if a minimum variance controller were implemented on the system described in equations (1)-(2). The variance of the closed-loop system can be calculated from the Markov parameters/impulse response of the closed-loop system and the minimum variance control variance expression for the process output is given by

$$var[y_t]_{MVC} = \sum_{i=0}^{\infty} \psi_i var[e_t] \psi_i^T \qquad (15)$$

Note that estimation of the interactor matrix is *not* required for obtaining the MVC-benchmark variance. However the above result requires the knowledge of $L_h = \begin{bmatrix} L_0 \\ ... \\ L_{N-1} \end{bmatrix}$, and hence it appears that estimation of the noise model in the Markov parameters model is necessary.

However, we will show that the estimation of $\begin{bmatrix} L_0 \\ ... \\ L_{N-1} \end{bmatrix}$ is *not* required. The closed loop noise model Markov parameters $L_h^{CL} = \begin{bmatrix} L_0^{CL} \\ ... \\ L_{N-1}^{CL} \end{bmatrix}$ (the vector of closed-loop noise model which can be estimated from the routine operating data) can be used in the place of $\begin{bmatrix} L_0 \\ ... \\ L_{N-1} \end{bmatrix}$ and we can still be able to obtain the MVC-benchmark variance, where

$$L_h^{CL} = (I + L_u L_c)^{-1} L_h \qquad (16)$$

and L_c represents the dynamic matrix containing the Markov parameters of the controller.

Lemma 1: Ψ *can be obtained using the vector of Markov parameters of the closed loop noise model,* L_h^{CL}*, in place of the* L_h *in equation (14) .*

Proof: The above statement is equivalent to saying that $(I - L_u L_u^{\dagger}) L_h$ and $(I - L_u L_u^{\dagger}) L_h^{CL}$ yield the same result. Now, $(I - L_u L_u^{\dagger}) L_h^{CL} = (I - L_u L_u^{\dagger})(I + L_u L_c)^{-1} L_h$. Therefore on observation, we need to show that

$$(I - L_u L_u^{\dagger}) = (I - L_u L_u^{\dagger})(I + L_u L_c)^{-1} \quad (17)$$

to prove the lemma, which is equivalent to showing

$$(I - L_u L_u^{\dagger})(I + L_u L_c) = (I - L_u L_u^{\dagger}) \quad (18)$$

Expanding the left hand side term in the above equation

$$(I - L_u L_u^{\dagger}) \;\; (I + L_u L_c) = I - L_u L_u^{\dagger} \quad (19)$$

The last equation follows since $L_u L_u^{\dagger} L_u = L_u$.

Lemma 1 is essentially the subspace version of the invariance property of the first few Markov parameters of the interactor-filtered noise model under the transfer function framework originally derived in Huang and Shah (1999)(Huang *et al.*, 1997). This invariance property has also been proved in (Ko and Edgar, 2001).

Hence the Markov parameters of the closed loop noise model can be used in place of Markov parameters of the open loop noise model and we can still get the same benchmark variance. Therefore, we need the subspace matrix L_u (which contains Markov parameters of the process and is estimated from data) for the calculation of the minimum variance control benchmark. The subspace matrix L_u can be estimated from closed loop data with set point excitation as explained in (Kadali and Huang, 2002). The Markov parameters of the closed loop noise model (or noise subspace matrix) can be easily estimated from the routine operating data (Kadali and Huang, 2002).

5. EQUIVALENCE OF SUBSPACE APPROACH AND THE CONVENTIONAL TRANSFER FUNCTION APPROACH IN OBTAINING THE MVC-BENCHMARK

In the transfer function approach,

$$D(z) = \left[D_d z^d + ... + D_1 z \right]$$

represents the interactor matrix for a process represented by the transfer function matrix $G(z^{-1}) = \left[G_0 + G_1 z^{-1} + G_2 z^{-2} + ... \right]$, then the condition for the interactor matrix from theorem 3.2.1 in (Huang and Shah, 1999) is

$$\lim_{z^{-1} \longrightarrow 0} DG = K \qquad (20)$$

where K is a full rank matrix. The above expression can be alternatively expressed as two matrix conditions

Condition-1

$$\begin{bmatrix} D_1 & ... & D_d \end{bmatrix} \begin{bmatrix} G_0 & ... & 0 \\ ... & ... & 0 \\ G_{d-1} & ... & G_0 \end{bmatrix} = \begin{bmatrix} 0 & ... & 0 \end{bmatrix} \quad (21)$$

Condition-2

$$\begin{bmatrix} D_1 & ... & D_d \end{bmatrix} \begin{bmatrix} G_1 \\ ... \\ G_d \end{bmatrix} = K \quad (22)$$

with $rank(K) = min\{m, l\}$.

We need to show that the coefficients obtained in the subspace approach are same as those obtained in the transfer function domain approach, i.e. the above two conditions are satisfied by using the matrix $\begin{pmatrix} I - L_u & L_u^\dagger \end{pmatrix}$. Therefore we have to prove the following theorem for the subspace approach:

Theorem 1: $\begin{pmatrix} I - L_u & L_u^\dagger \end{pmatrix}$ *contains interactor matrix for the process. An interactor matrix can be constructed directly from this expression. The subspace approach for the calculation of the minimum variance control benchmark is equivalent to that of the conventional transfer function approach.*

Proof:

$$\begin{pmatrix} I - L_u & L_u^\dagger \end{pmatrix} \begin{bmatrix} G_0 & 0 & ... & 0 \\ G_1 & G_0 & 0 & ... \\ ... & ... & ... & ... \end{bmatrix} = \begin{pmatrix} I - L_u & L_u^\dagger \end{pmatrix} L_u$$
$$= L_u - L_u \ L_u^\dagger L_u = 0 \quad (23)$$

since $L_u \ L_u^\dagger L_u = L_u$. From the above equation condition-1 expressed in equation (21) is satisfied.

Next, consider the transformed Markov parameter matrix $\begin{bmatrix} \hat{G}_1 \\ ... \\ \hat{G}_d \end{bmatrix} = (I - L_u L_u^\dagger) \begin{bmatrix} G_1 \\ ... \\ G_d \end{bmatrix}$. Note that the matrices L_u and $\begin{bmatrix} G_1 \\ ... \\ G_d \end{bmatrix}$ are *essentially disjoint* (see appendix A). Following the corollary 17.2.10 in ref.(Harville, 1997) :

$$rank \begin{bmatrix} \hat{G}_1 \\ ... \\ \hat{G}_d \end{bmatrix} = rank \begin{bmatrix} G_1 \\ ... \\ G_d \end{bmatrix} \quad (24)$$

Now let

$$K = \hat{G}_1 + ... + \hat{G}_d = \begin{bmatrix} I_m & ... & I_m \end{bmatrix} \begin{bmatrix} \hat{G}_1 \\ ... \\ \hat{G}_d \end{bmatrix} \quad (25)$$

The matrix $\begin{bmatrix} I_m & ... & I_m \end{bmatrix}$ is $(m \times dm)$ dimensional with rank m. Consider $A = \begin{bmatrix} I_m & ... & I_m \end{bmatrix}_{m \times dm}$ and $B = \begin{bmatrix} \hat{G}_1 \\ ... \\ \hat{G}_d \end{bmatrix}_{dm \times l}$. Using the corollary 17.5.2 from (Harville, 1997) we can write

$$rank\{K\} = rank[A] + rank[B] - (dm)$$
$$+rank\left[(I_{dm} - BB^\dagger)(I_{dm} - A^\dagger A)\right] \quad (26)$$

We can expand

$$(I - BB^\dagger)(I - A^\dagger A)$$
$$= (I - A^\dagger A) - B^\dagger B(I - A^\dagger A) \quad (27)$$

For using the item (3) in appendix B, we take,

$$R = (I - A^\dagger A); \ \ S = -B; \ \ T = B^\dagger; \ \ U = (I - A^\dagger A)$$

Using equations (26) and (B.2), we write,

$$rank[K] = rank[A] + rank[B] - dm - rank[B^\dagger]$$
$$+rank[(I - A^\dagger A)] + rank[(AB)^\dagger (AB)B^\dagger]$$
$$= rank[(AB)^\dagger (AB)B^\dagger] \quad (28)$$

In the above equation we used $rank[B] = rank[B^\dagger]$ and $rank[(I - A^\dagger A)] = (d-1)m$. Consider the two cases,

(i) $m \geq l$: In this case $(AB)^\dagger (AB) = I_l$. Therefore $rank[(AB)^\dagger (AB)B^\dagger] = rank[B^\dagger] = l$ and $rank[K] = l$.

(ii) $m < l$: In this case $rank[(AB)^\dagger (AB)] = m$. Since B is a full rank matrix $rank[(AB)^\dagger (AB)B^\dagger] = m$. Therefore $rank[K] = m$.

Hence K is a full rank matrix and condition-2 expressed in equation (22) is satisfied. Hence the theorem is proved.

To summarize, the matrix $\begin{pmatrix} I - L_u & L_u^\dagger \end{pmatrix}$ performs the same function as an interactor matrix in the transfer function domain. But the calculation of interactor matrix is not required in deriving the MVC-benchmark variance of the process output for controller performance analysis. Therefore *a priori* knowledge of only the subspace matrix L_u or equivalently the first 'd' process Markov parameters is the requirement for obtaining the MVC-benchmark, the same conclusion as obtained in the previous literature, but expressed here in a much more simplified notation..

6. CONCLUSIONS

Calculation of the multivariate performance index without using the interactor matrix is an impor-

tant step toward practical application of multivariate performance assessment techniques. It is shown in this paper the design of multivariate minimum variance controller can be done using subspace matrices. Using the subspace matrices the MVC-benchmark variance for the process outputs is obtained from closed loop data without having to first calculate the unitary interactor matrix or knowing the first few Markov parameters of the noise model. The equivalence of the subspace approach to the conventional transfer function approach for obtaining the MVC-benchmark variance is also proved.

REFERENCES

Astrom, K.J. and B. Wittenmark (1984). *Computer Control Systems: Theory and Design*. Prentice-Hall.

Harris, T. (1989). Assessment of closed loop performance. *Can. J. Chem. Eng.* **67**, 856–861.

Harris, T.J., F. Boudreau and J.F. MacGregor (1996). Performance assessment of multivariable feedback controllers. *Automatica* **32**(11), 1505–1518.

Harville, D.A. (1997). *Matrix Algebra From a Statistician's Perspective*. Springer-Verlag, New York.

Huang, B. and S.L. Shah (1999). *Control Loop Performance Assessment: Theory and Applications*. Springer Verlag, London.

Huang, B., S.L. Shah and K.Y. Kwok (1997). Good, bad or optimal? performance assessment of multivariable processes. *Automatica* **33**(6), 1175–1183.

Kadali, R. and B. Huang (2002). Estimation of dynamic matrix and noise model for model predictive control using closed loop data. *Industrial and Engineering Chemistry Research* **41**(4), 842–852.

Ko, B.S. and T.F. Edgar (2001). Performance assessment of multivariate feedback control systems. *Automatica* **37**(5), 899–905.

McNabb, C.A.R. and J.S. Qin (2001). MIMO control performance monitoring based on subspace projections. In: *AIChE Annual Conference, Reno*.

Overschee, P. Van and B. De Moor (1994). N4SID: Subspace algorithm for the identification of combined deterministic-stochastic systems. *Automatica* **30(1)**, 75–93.

Overschee, P. Van and B. De Moor (1995). A unifying theorem for three subspace system identification algorithms. *Automatica* **31(12)**, 1877–1883.

Overschee, P. Van and B. De Moor (1996). *Subspace Identification for Linear Systems: Theory implementation applications*. Kluwer Academic Publishers.

Appendix A. ESSENTIALLY DISJOINT CONDITION

From (Harville, 1997)

Lemma 17.1.4. Let U and V represent subspaces of $R^{m\times n}$, then

(1) U and V are essentially disjoint if and only if, for matrices $\mathbf{U}\in U$ and $\mathbf{V}\in V$, the only solution to the matrix equation

$$\mathbf{U}+\mathbf{V}=0 \tag{A.1}$$

is $\mathbf{U}=\mathbf{V}=0$; and

(2) U and V are essentially disjoint if and only if, for every non-null matrix $\mathbf{U}\in U$ and every non-null matrix $\mathbf{V}\in V$, $\mathbf{U}$ and $\mathbf{V}$ are linearly independent.

We assume that the process transfer function $G(z^{-1})$ is full rank with proper and stable transfer functions. Therefore, the matrices $L_u = \begin{bmatrix} G_0 & 0 & ... \\ G_1 & G_0 & ... \\ ... & ... & ... \end{bmatrix}$ and $\begin{bmatrix} G_1 \\ G_2 \\ ... \end{bmatrix}$ are essentially disjoint.

Appendix B. COROLLARIES

(1) Corollary 17.5.2 Let A represent an $m\times n$ matrix and B an $n\times p$ matrix. Then,

$$\begin{aligned} rank(AB) &= rank(A)+rank(B)-n \\ &\quad +rank\left[(I-BB^{\dagger})(I-A^{\dagger}A)\right] \end{aligned} \tag{B.1}$$

(2) Corollary 17.2.10 Let A represent an $m\times n$ matrix, B an $m\times p$ matrix. Then $\text{rank}\big[(I-AA^{\dagger})B\big] = \text{rank}\,(B)$ if and only if $\text{C}(A)$ and $\text{C}(B)$ are essentially disjoint.

(3) From chapter 18 Let R represent an $n\times q$ matrix, S an $n\times m$ matrix, T an $m\times p$ matrix, and U a $p\times q$ matrix. Then,

$$\begin{aligned} rank(R+STU) &= rank(R)+rank(Q)+rank(M) \\ &\quad +rank(N)-rank(T) \\ &\quad +rank\left[(I-MM^{\dagger})XQ^{\dagger}Y(I-N^{\dagger}N)\right] \end{aligned} \tag{B.2}$$

where $E_R = I-RR^{\dagger}$; $F_R = I-R^{\dagger}R$; $X = E_RST$; $Y = TUF_R$; $M = X(I-Q^{\dagger}Q)$ and $N = (I-QQ^{\dagger})Y$. Refer to (Harville, 1997) for proofs of the above corollaries.