Adaptive Process Monitoring via Multichannel EIV Lattice Filters

Weihua Li, Abhishek Bhargava, and Sirish L. Shah
Dept. of Chemical and Materials Engineering, University of Alberta, Edmonton, AB, T6G 2G6, Canada

On line monitoring of multivariable processes is crucial to operational safety and product quality. For this, multivariable statistical analysis methods, such as principal-component analysis (PCA), partial least squares, and canonical variate analysis have been widely applied. However, few recursive monitoring techniques have been developed for fully dynamic and time-varying processes. Recursive PCA has been successfully applied to monitor static time-varying processes, does not work for fully dynamic processes. Dynamic PCA has been developed, but its recursive variant is not available. Many processes operate in dynamic states and are often time-varying, and the time-varying property includes the variation of parameters and of process structure, e.g., the change of model order. A novel approach to the adaptive monitoring of multivariate dynamic and time-varying processes by the recursive multichannel instrumental variable (IV) lattice filters was developed using the errors-in-variables (EIV) state space model to represent a dynamic process. To show the relationship between EIV state-space representation of the process and a multichannel IV lattice filter, the lattice filter was used to generate a residual vector for process monitoring. By using lattice filter's ability of recursively updating the process model both in time and order, a real time, on-line algorithm was used to update the residual vector with newly sampled process data, including a practical approach to recursive determination of the process model order. Based on the residual vector, the Hotelling T^2 statistic and the associated confidence limits are used as the monitoring index. The proposed scheme was evaluated on a simulation example and a pilot plant to support the theoretical results.

Introduction

On line monitoring of multivariable processes is crucial to operational safety and product quality. Multivariable statistical analysis methods, such as principal-component analysis (PCA), partial least squares (PLS), canonical variate analysis (CVA), and other state-space methods have been widely applied to chemical processes for monitoring and diagnosis (MacGregor, 1989; Kresta et al., 1991; Piovoso et al., 1992; Wise et al., 1995; Kourti and MacGregor, 1995; Larimore, 1983, 1997; Negiz and Cinar, 1997, 1998). However, for dynamic and time-varying processes, few monitoring techniques are available. Recursive PCA (Li et al., 2000) has been proposed and successfully applied to quasi-steady process monitoring, but it does not work for fully dynamic processes. Dynamic PCA has also been developed for processes diagnosis (Ku et al., 1995); however, its recursive variant is not available yet. The subspace methods of identification (SMI) for the errors-in-variables (EIV) state-space model has been proposed by Chou and Verhaegen (1997), and has been directly applied to residual generation for process monitoring and fault detection (Qin and Li, 2001; Li and Shah, 2002). More recently, the SMI has been improved to be recursive in time by Gustafsson et al., (2000), but the improved SMI is still not recursive in order.

Chemical processes often operate in dynamic state and demonstrate slowly time-varying behaviors due to, for example, catalyst deactivation, equipment aging, sensor and process drifting, and preventive maintenance and cleaning. The time-varying property includes not only the variation in parameters but also the variation in process structure, which can be represented by the change in the process order. So far, to our best knowledge, little has been reported on the

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

development of adaptive monitoring techniques for time- and order-varying multivariate dynamic processes.

In this article, we propose a novel approach to adaptive monitoring of dynamic time-varying processes via the multichannel instrumental variable (IV) lattice filters. We use the errors-in-variables (EIV) state-space model to represent a multivariate dynamic process in the presence of process noise, plus measurement noise in the inputs and outputs. By showing the relationship between the process EIV state-space representation and a multichannel lattice filter, we propose the lattice filter to generate a residual vector for process monitoring.

Lattice filters have been extensively applied to signal processing and system identification (Lee et al., 1981; Friedlander, 1982; Kummert et al., 1992). Their main advantage is that they are recursive both in time and order. In addition, they have many other useful properties, such as modular locally interconnected structure, low sensitivity of their coefficients to numerical perturbations, and the algorithms' relative indifference to the eigenvalue spread of the process data covariance matrix, and the attribute of allowing fast convergence and good tracking of time-varying process variables (Honig and Messerschmitt, 1981; Lev-Ari et al., 1984).

Unfortunately, existing conventional lattice filters fail to give a consistent estimate of the multivariate process model in the EIV case, that is, parameters estimated by the conventional lattice filters are biased from their true values. Friedlander (1983) proposes an IV lattice filter, which is able to give a consistent estimate of the system model only when the output of the system is corrupted, but is unable to do so in the EIV case. In addition, Friedlander's scheme does not work for multivariate systems.

When a conventional lattice filter is applied to generate residuals for monitoring a system with errors in the inputs and outputs, the generated residuals will be affected by the biases in the estimated parameters of the system, even though the system under consideration is operating normally. As a consequence, false alarms will result. We have developed an IV multichannel lattice filter (Li and Shah, 2000) that can give a consistent estimate of the dynamic process model while preserving all the advantages that the conventional lattice filters have. In this article, we propose an efficient algorithm to update the residual vector in time and order simultaneously by the use of the IV multichannel lattice filter. Further, we employ the computed residual vector for constructing the Hotelling T^2 statistic as a monitoring index. The proposed approach is evaluated by monitoring a simulated process and a time-varying pilot plant, respectively.

This article is organized as follows. The following section outlines the problem formulation. In the third section, we review the concept of the IV methods, establish the unified signal space for inputs and outputs, introduce new notations for the development of the IV lattice filter, and show the equivalence between the EIV state-space model and the IV multichannel lattice filter with regard to the generation of the residual vector for process monitoring. The complete algorithm for the recursive update of the residual vector in time and order is given in the Appendices, including the initialization. The fourth section investigates the application of the IV lattice filter for adaptive process monitoring, where treatment of certain practical issues is also considered. Case studies for evaluation of the proposed algorithms are documented in the fifth section, where we also compare the proposed monitoring approach to a constant model-based monitoring approach. The article ends with concluding remarks in the final section.

Problem Formulation

Assume that the normal behavior of the process under consideration be represented by the unknown discrete state-space model, as in Chou and Verhaegen (1997), as follows

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\tilde{\boldsymbol{u}}(k) + \boldsymbol{K}\boldsymbol{p}(k) \\ \tilde{\boldsymbol{y}}(k) &= \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\tilde{\boldsymbol{u}}(k) \end{aligned} \quad (1)$$

where $\tilde{\boldsymbol{u}}(k) \in \Re^l$ and $\tilde{\boldsymbol{y}}(k) \in \Re^m$ are noise-free inputs and outputs, respectively, and $\boldsymbol{x}(k) \in \Re^n$ is the state vector. In addition, $\boldsymbol{p}(k) \in \Re^d$ is the process noise, and $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$, and $\boldsymbol{K}$ are system and noise gain matrices with compatible dimensions, respectively.

We assume that the observed input $\boldsymbol{u}(k)$ and output $\boldsymbol{y}(k)$ are corrupted by zero-mean Gaussian distributed noise vectors $\boldsymbol{v}(k)$ and $\boldsymbol{o}(k)$, respectively, that is

$$\begin{aligned} \boldsymbol{u}(k) &= \tilde{\boldsymbol{u}}(k) + \mathrm{v}(k) \\ \boldsymbol{y}(k) &= \tilde{\boldsymbol{y}}(k) + \boldsymbol{o}(k) \end{aligned} \quad (2)$$

This is referred to as the case of EIV (Chou and Verhaegen, 1997). We assume that (i) $\boldsymbol{v}(k)$ and $\boldsymbol{o}(k)$ are mutually independent; and (ii) $\boldsymbol{v}(k)$ and $\boldsymbol{o}(k)$ are independent of $\tilde{\boldsymbol{u}}(k)$ and $\tilde{\boldsymbol{y}}(k)$, respectively.

Substituting Eq. 2 into Eq. 1, and performing an algebraic manipulation, shows

$$\boldsymbol{y}_n(k) = \boldsymbol{\Gamma}_n \boldsymbol{x}(k-n) + \boldsymbol{H}_n \boldsymbol{u}_n(k) - \boldsymbol{H}_n \mathrm{v}_n(k) + \boldsymbol{G}_n \boldsymbol{p}_n(k) + \boldsymbol{o}_n(k) \quad (3)$$

where

$$\boldsymbol{y}_n(k) = \left[\boldsymbol{y}^T(k) \cdots \boldsymbol{y}^T(k-n) \right]^T \in \Re^{m_n}$$

is the augmented output vector,

$$\boldsymbol{\Gamma}_n = \left[(\boldsymbol{C}\boldsymbol{A}^n)^T \cdots \boldsymbol{C}^T \right]^T \in \Re^{m_n \times n}$$

is the extended observability matrix with rank n

$$\boldsymbol{H}_n = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{C}\boldsymbol{B} & \cdots & \cdots & \boldsymbol{C}\boldsymbol{A}^{n-1}\boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{D} & \boldsymbol{C}\boldsymbol{B} & \cdots & \vdots \\ \vdots & & \ddots & \ddots & \\ \vdots & & & & \boldsymbol{C}\boldsymbol{B} \\ \boldsymbol{0} & \cdots & & & \boldsymbol{D} \end{bmatrix} \in \Re^{m_n \times l_n}$$

and

$$G_n = \begin{bmatrix} \mathbf{0} & CK & \cdots & \cdots & CA^{n-1}K \\ & \mathbf{0} & CK & \cdots & \vdots \\ \vdots & & \ddots & \ddots & \\ \vdots & & & & CK \\ \mathbf{0} & \cdots & & & \mathbf{0} \end{bmatrix} \in \Re^{m_n \times d_n}$$

are two upper triangular block Toeplitz matrices with $m_n = (n+1)m$, $l_n = (n+1)l$, and $d_n = (n+1)d$. The vectors $\boldsymbol{v}_n(k) \in \Re^{l_n}$, $\boldsymbol{o}_n(k) \in \Re^{m_n}$, $\boldsymbol{p}_n(k) \in \Re^{d_n}$, and $\boldsymbol{u}_n(k) \in \Re^{l_n}$ are defined in the same format as $\boldsymbol{y}_n(k)$.

Assume that the characteristic polynomial of $A - \chi I$ is as follows:

$$\det(A - \chi I) = \chi^n + \alpha_1 \chi^{n-1} + \cdots + \alpha_{n-1}\chi + \alpha_n = 0.$$

In accordance with the Cayley-Hamilton theorem (Kailath, 1980), premultiplying both sides of Eq. 3 by the following matrix

$$\left(\Gamma_n^{\perp}\right)^T = [I_m \ \alpha_1 I_m \cdots \alpha_n I_m] \in \Re^{m \times m_n}$$

gives

$$\begin{aligned} \boldsymbol{y}(k) + \alpha_1 \boldsymbol{y}(k-1) + \cdots + \alpha_{n-1}\boldsymbol{y}(k-n+1) + \alpha_n \boldsymbol{y}(k-n) \\ = \left(\Gamma_n^{\perp}\right)^T H_n \boldsymbol{u}_n(k) + \left(\Gamma_n^{\perp}\right)^T \boldsymbol{o}_n(k) - \left(\Gamma_n^{\perp}\right)^T H_n \boldsymbol{v}_n(k) \\ + \left(\Gamma_n^{\perp}\right)^T G_n \boldsymbol{p}_n(k) \end{aligned} \quad (4)$$

where note that $\Gamma_n^{\perp}$ is part of the null space of Γ_n.

Clearly, Eq. 4 is a typical autoregressive moving-average exogenous (ARMAX) process in terms of the definition given by Ljung (1987). Denoting

$$\left(\Gamma_n^{\perp}\right)^T H_n = \left[D \quad CB + \alpha_1 D \quad CAB + \alpha_1 CB + \alpha_2 D \cdots \sum_{j=0}^{n-1} \alpha_j CA^{n-1-j}B + \alpha_n D \right] \equiv \left[L^{(0)} \ L^{(1)} \cdots L^{(n)} \right] \in \Re^{m \times l_n}$$

and

$$\left(\Gamma_n^{\perp}\right)^T G_n = \left[\mathbf{0} \quad CK \quad CAK + \alpha_1 CK \cdots \sum_{j=0}^{n-1} \alpha_j CA^{n-1-j}K \right] \equiv \left[\mathbf{0} \ \Psi^{(1)} \cdots \Psi^{(n)} \right] \in \Re^{m \times l_n}$$

we rewrite Eq. 4 as

$$\begin{aligned} \boldsymbol{e}_n(k) &\equiv \boldsymbol{y}(k) + \sum_{s=1}^{n} \alpha_s \boldsymbol{y}(k-s) - \sum_{s=0}^{n} L^{(s)} \boldsymbol{u}(k-\omega) \\ &= \boldsymbol{o}(k) + \sum_{s=1}^{n} \alpha_s \boldsymbol{o}(k-s) - \sum_{s=0}^{n} L^{(s)} \boldsymbol{v}(k-s) \\ &\quad + \sum_{s=1}^{n} \Psi^{(s)} \boldsymbol{p}(k-s) \end{aligned} \quad (5)$$

where $\boldsymbol{e}_n(k)$ is defined as the residual vector and will be used for process monitoring. Note that in Eq. 5 the first line on the righthand side (RHS) shows how $\boldsymbol{e}_n(k)$ is computed from the sampled data $\{\boldsymbol{y}(k) \ \boldsymbol{u}(k)\}$ and their time-lagged values $\{\boldsymbol{y}(k-1) \ \boldsymbol{u}(k-1) \cdots \boldsymbol{y}(k-n) \ \boldsymbol{u}(k-n)\}$; but the second line on the RHS reveals the internal relation bewteen $\boldsymbol{e}_n(k)$, $[\boldsymbol{o}^T(k) \ \boldsymbol{v}^T(k) \ \boldsymbol{p}^T(k)]$, and the time-lagged values of $[\boldsymbol{o}^T(k) \ \boldsymbol{v}^T(k) \ \boldsymbol{p}^T(k)]$.

Apparently, Eq. 5 can be decomposed into m independent equations, for example, the ith residual equation for $i = 1 \cdots m$ is

$$\begin{aligned} e_n^i(k) = y_i(k) - \left(\boldsymbol{\theta}_n^i\right)^T \boldsymbol{\xi}_n^i(k) = \sum_{s=0}^{n} \alpha_s o_i(k-s) \\ - \sum_{s=0}^{n} \boldsymbol{l}^{(s)}(i,:)\boldsymbol{v}(k-s) + \sum_{s=1}^{n} \boldsymbol{\psi}^{(s)}(i,:)\boldsymbol{p}(k-s) \end{aligned} \quad (6)$$

where $\alpha_0 = 1$; $y_i(k)$ and $o_i(k)$ are the ith element of the vectors $\boldsymbol{y}(k)$ and $\boldsymbol{o}(k)$, respectively; in addition

$$\boldsymbol{\xi}_n^i(k) = \left[\boldsymbol{u}^T(k) \, y_i(k-1) \, \boldsymbol{u}^T(k-1) \cdots y_i(k-n) \, \boldsymbol{u}^T(k-n) \right]^T \in \Re^{l_n + n} \quad (7)$$

and $\boldsymbol{\theta}_n^i = [\boldsymbol{l}^{(0)}(i,:) \ -\alpha_1 \boldsymbol{l}^{(1)}(i,:) \ -\alpha_2 \boldsymbol{l}^{(2)}(i,:) \cdots -\alpha_n \boldsymbol{l}^{(n)}(i,:)]^T \in \Re^{(l_n+n)}$, with $\boldsymbol{l}^{(\cdot)}(i,:)$ and $\boldsymbol{\psi}^{(\cdot)}(i,:)$ symbolizing the ith row of the matrices $L^{(\cdot)}$ and $\boldsymbol{\psi}^{(\cdot)}$, respectively.

Using Eq. 6, Eq. 5 can be rewritten into

$$\begin{aligned} \boldsymbol{e}_n(k) \equiv \begin{bmatrix} s_n^1(k) \\ \vdots \\ e_n^m(k) \end{bmatrix} = \boldsymbol{y}(k) - \begin{bmatrix} \left(\boldsymbol{\theta}_n^1\right)^T \boldsymbol{\xi}_n^1(k) \\ \vdots \\ \left(\boldsymbol{\theta}_n^m\right)^T \boldsymbol{\xi}_n^m(k) \end{bmatrix} \\ = \boldsymbol{o}(k) - L^{(0)}\boldsymbol{v}(k) + \sum_{s=1}^{n} \left[\alpha_s \boldsymbol{o}(k-s) \right. \\ \left. - L^{(s)}\boldsymbol{v}(k-s) + \Psi^{(s)}\boldsymbol{p}(k-s) \right] \end{aligned} \quad (8)$$

The following remarks are in order, on the basis of the examination of the preceding residual vector:

1. $\boldsymbol{e}_n(k)$ consists of m elements $\{e_n^1(k) \cdots e_n^m(k)\}$. To generate $\boldsymbol{e}_n(k)$, an m-channel lattice filter is needed. Note that a lattice filter is said to be *multichannel* if the residual $\boldsymbol{e}_n(k)$ contains more than one element in accordance with the definition established by Proakis and Manolakis (1996). However, since each element of $\boldsymbol{e}_n(k)$ is independent of the other, in practice we only have to derive the algorithm of the lattice filter for the ith channel with $i \in [1, \cdots, m]$.
2. The model parameters of $\boldsymbol{e}_n(k)$ are

$$\Theta_n \equiv \begin{bmatrix} \left(\boldsymbol{\theta}_n^1\right)^T \\ \vdots \\ \left(\boldsymbol{\theta}_n^m\right)^T \end{bmatrix} \in \Re^{m \times (l_n + n)}$$

However, we propose to generate $\boldsymbol{e}_n(k)$ recursively in time and order by using the sampled data $[\boldsymbol{y}_n^T(k) \ \boldsymbol{u}_n^T(k)]^T$ without estimating the model parameters explicitly.

3. In the ideal case that the process operates normally and there is no change in the process, $\boldsymbol{e}_n(k) = \boldsymbol{e}_n^*(k)$, where

$$e_n^*(k) = o(k) - L^{(0)}v(k) + \sum_{s=1}^{n}\left[\alpha_s o(k-s) - L^{(s)}v(k-s) + \psi^{(s)}p(k-s)\right]$$

is a vector MA process of the noise vector $[o^T(k) v(k)^T p^T(k)]^T$, with the covariance matrix as follows:

$$R_n^{e*}(k) = \sum_{s=0}^{n}\alpha_s^2 E\{o(k)[o(k)]^T\} + \sum_{s=0}^{n} L^{(s)}E\{v(k)[v(k)]^T\}(L^{(s)})^T + \sum_{s=1}^{n}\psi^{(s)}E\{p(k)[p(k)]^T\}(\psi^{(s)})^T \in \Re^{m\times m}$$

Once there are slowly time-varying changes in the process, the lattice filter can track such changes recursively. As a consequence, the residual vector $e_n(k)$ still can be zero mean, but the covariance matrix R_n^e can be different from R_n^{*e}. The recursive update of R_n^e is also proposed in this article.

4. Any undesired changes in the parameters and structure of the process model will manifest themselves as changes in the mean of $e_n(k)$. Therefore, process monitoring is equivalent to checking if the mean of the recursively updated $e_n(k)$ deviates from zero. This can be done by checking if the following newly defined variable $e_n^T(k)(R_n^e)^{-1}e_n(k)$ follows a chi-square distribution, where as mentioned earlier, R_n^e has to be recursively updated.

Instrumental Variable Lattice Filter

The lattice filter was originally proposed by Itakura and Saito (1971) for speech analysis and synthesis. Later, for a variety of reasons as outlined in the Introduction, it found wide applications in adaptive signal processing (Friedlander, 1982; Lev-Ari et al., 1984), and system identification (Ljung, 1987; Jabbari and Gibson, 1988).

The original lattice filters could only be used to identify the AR models. Then Lee et al. (1982), Jabbari and Gibson (1988), and Kummert et al. (1992) extended them to the identification of ARMA models. Nevertheless, the dynamic process under consideration in this article is a typical ARMAX process.

In this section, we start by showing that the proposed IV lattice filter can give a consistent estimate of the parameters in the considered ARMAX process. Then, we give the complete IV lattice filter algorithm, including order and time-recursion equations with initial conditions. Furthermore, we will demonstrate the use of the lattice filter for the generation of the residual vector $e_n(k)$. A detailed derivation of the lattice filters is lengthy and beyond the scope of this article. The interested readers are referred to Li and Shah (2000).

Instrumental variable methods

It is well-known (Ljung, 1987) that the least-square estimate of θ_n^i is

$$\hat{\theta}_{n,ls}^i = \left\{\frac{1}{N}\sum_{k=1}^{N}\xi_n^i(k)[\xi_n^i(k)]^T\right\}^{-1}\frac{1}{N}\sum_{k=1}^{N}\xi_n^i(k)y_i(k) \quad (9)$$

where N is the number of data samples. Using Eq. 6 in Eq. 9 leads to

$$\hat{\theta}_{n,ls}^i = \theta_n^i + \left\{\frac{1}{N}\sum_{k=1}^{N}\xi_n^i(k)[\xi_n^i(k)]^T\right\}^{-1}\frac{1}{N}\sum_{k=1}^{N}\xi_n^i(k) \times\left[\sum_{s=0}^{n}\alpha_s o_i(k-s) - \sum_{s=0}^{n} l^{(s)}(i,:)v(k-s) + \sum_{s=1}^{n}\psi^{(s)}(i,:)p(k-s)\right] \quad (10)$$

Consider the fact that the noise term

$$\sum_{s=0}^{n}\alpha_s o_i(k-s) - \sum_{s=0}^{n} l^{(s)}(i,:)v(k-s) + \sum_{s=1}^{n}\psi^{(s)}(i,:)p(k-s)$$

is correlated with $\xi_n^i(k)$; then we can show that

$$\lim_{N\to\infty}\hat{\theta}_{n,ls}^i \neq \theta_n^i$$

To remove the effect of the noise term on the estimation of θ_n^i, we will incorporate the IV method with the lattice filter. For instance, we can choose an instrumental variable regressor

$$\rho_n^i(k) = [u^T(k-\mu)\ y_i(k-1-\mu)\ u^T(k-1-\mu)\cdots y_i(k-n-\mu)\ u(k-n-\mu)]^T \quad (11)$$

such that

$$\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}\rho_n^i(k)[\xi_n^i(k)]^T$$

has full rank, and

$$\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}\rho_n^i(k)\left[\sum_{s=0}^{n}\alpha_s o_i(k-s) - \sum_{s=0}^{n} l^{(s)}(i,:)v(k-s) + \sum_{s=1}^{n}\Psi^{(s)}(i,:)p(k-s)\right] = 0$$

Consequently, the IV estimate $\hat{\theta}_{n,iv}^i$ of θ_n^i will be

$$\hat{\theta}_{n,iv}^i = \left\{\frac{1}{N}\sum_{k=1}^{N}\rho_n^i(k)[\xi_n^i(k)]^T\right\}^{-1}\frac{1}{N}\sum_{k=1}^{N}\rho_n^i(k)y_i(k)$$
$$= \theta_n^i + \left\{\frac{1}{N}\sum_{k=1}^{N}\rho_n^i(k)[\xi_n^i(k)]^T\right\}^{-1}\frac{1}{N}\sum_{k=1}^{N}\rho_n^i(k) \times\left[\sum_{s=0}^{n}\alpha_s o_i(k-s) - \sum_{s=0}^{n} l^{(s)}(i,:)v(k-s) + \sum_{s=1}^{n}\psi^{(s)}(i,:)p(k-s)\right] \quad (12)$$

which tends to $\boldsymbol{\theta}_n^i$ as N tends to infinity. It will be shown that there is a one-to-one correspondence between $\hat{\boldsymbol{\theta}}_n^i$ and the signals processed by the proposed IV lattice filter.

Unified signal subspace

It follows directly from Eqs. 6 that $e_n^i(k)$ is a linear combination of the following signals

$$\{y_i(k)\ \boldsymbol{u}(k)\cdots y_i(k-1)\ \boldsymbol{u}(k-1)\cdots y_i(k-n)\ \boldsymbol{u}(k-n)\}$$

The key for the multichannel lattice filters to generate the residual $e_n^i(k)$ is to establish a unified notation to represent all the input and output signals so that they can be handled in a unified frame work. We define the following unified signal (Kummert et al., 1992):

$$z^i(kM-s)=\begin{cases} y_i(k), & s=1 \\ u_{s-1}(k), & s=2\cdots M, \text{ where } M=l+1 \end{cases} \quad (13)$$

where $u_j(k)$ for $j=1\cdots l$ stands for the jth element of the input vector $\boldsymbol{u}(k)$. The establishment of Eq. 13 results in the following unique correspondence from an input or an output to the unified signal $z^i(tM-s)$, $\forall\ \{s=1\cdots M,\ t=k\cdots k-n\}$, for example,

$$\begin{bmatrix} z^i(kM-1) \\ z^i(kM-2) \\ \vdots \\ z^i(kM-M) \\ z^i((k-1)M-1) \\ z^i((k-1)M-2) \\ \vdots \\ z^i((k-n)M-M) \\ \vdots \\ z^i((k-n)M-1) \\ z^i((k-n)M-2) \\ \vdots \\ z^i((k-n)M-M) \end{bmatrix} \equiv \begin{bmatrix} y_i(k) \\ u_1(k) \\ \vdots \\ u_l(k) \\ y_i(k-1) \\ u_1(k-1) \\ \vdots \\ u_l(k-1) \\ \vdots \\ y_i(k-n) \\ u_1(k-n) \\ \vdots \\ u_l(k-n) \end{bmatrix} \quad (14)$$

We define a signal vector with infinite length

$$\boldsymbol{z}_{kM-s}^i=\left[z^i(kM-s)\ z^i((k-1)\ M-s)\cdots z^i(1M-s)\ 0\cdots 0\right]^T$$

which belongs to the Hilbert space, for example, the vector $\boldsymbol{z}_{kM-s}^i$ has real elements only, and for $\boldsymbol{z}_{kM-s}^i$ and $\boldsymbol{z}_{(k-\mu)M-s}^i$, their inner product is

$$<\boldsymbol{z}_{kM-s}^i,\ \boldsymbol{z}_{(k-\mu)M-s}^i> = \times\sum_{t=1}^{k}\lambda^{k-t}z^i(tM-s)z^i\left[(t-\mu)M-s\right]<\infty$$

where $s=1\cdots M$, $z^i(t)=0$ for $t<0$, and $0<\lambda\le 1$ is a forgetting factor.

We define a subspace

$$\boldsymbol{h}_{s,\eta}^i(k)=\mathrm{Span}\left\{\boldsymbol{z}_{kM-s-1}^i,\boldsymbol{z}_{kMs-2}^i,\cdots,\ \boldsymbol{z}_{kM-s-\eta}^i\right\}$$

which is spanned by columns of the following matrix

$$\boldsymbol{z}_{kM-s-1:kM-s-\eta}^i=\left[\boldsymbol{z}_{kM-s-1}^i\ \boldsymbol{z}_{kM-s-2}^i\ \cdots\ \boldsymbol{z}_{kM-s-\eta}^i\right] \in\Re^{\infty\times\eta},\ \eta=1\cdots M_n-1$$

where $M_n=M(n+1)$. Further, we define

$$\boldsymbol{f}_{s,\eta}^i(k)=\boldsymbol{z}_{kM-s}^i-\boldsymbol{z}_{kM-s}^i|h_{s,\eta}^i(k),\quad \eta=1\cdots M_n-1;\ s=1\cdots M \quad (15)$$

as the forward error vector of the IV lattice filter, which is the difference between the vector $\boldsymbol{z}_{kM-s}^i$ and its nonsymmetric projection onto the subspace $\boldsymbol{h}_{s,\eta}^i(k)$. As will be shown later, while the conventional least-square lattice filter can be derived by performing a symmetric projection onto the subspace $\boldsymbol{h}_{s,\eta}^i(k)$, to derive the IV lattice filter, we must perform a nonsymmetric projection on the subspace.

The nonsymmetric projection of $\boldsymbol{z}_{kM-s}^i$ onto the subspace $\boldsymbol{h}_{s,\eta}^i(k)$ is

$$\boldsymbol{z}_{kM-s}^i|_{\boldsymbol{h}_{s,\eta}^i(k)}=\boldsymbol{z}_{kM-s-1:kM-s-\eta}^i\left[\left(\boldsymbol{z}_{(k-\mu)M-s-1:(k-\mu)M-s-n}^i\right)^T \times\boldsymbol{z}_{kM-s-1:kM-s-\eta}^i\right]^{-1}\left(\boldsymbol{z}_{(k-\mu)M-s-1:(k-\mu)M-s-\eta}^i\right)^T\boldsymbol{z}_{kM-s}^i \quad (16)$$

Geometrically, $\boldsymbol{f}_{s,\eta}^i(k)$ is perpendicular to the subspace $\boldsymbol{h}_{s,\eta}^i(k-\mu)$, where μ is a positive integer. Note that the nonsymmetric projection just defined is reduced to the conventional symmetric projection when $\mu=0$.

Denote the first element of $\boldsymbol{f}_{s,\eta}^i(k)$ by

$$\boldsymbol{\epsilon}_{s,\eta}^i(k)\equiv\boldsymbol{\phi}^T\boldsymbol{f}_{s,\eta}^i(k) \quad (17)$$

with $\boldsymbol{\phi}=[1\ 0\cdots 0]^T\in\Re^{\infty}$, then from Eqs. 15, 16, and 17, we have

$$\boldsymbol{\epsilon}_{1,M_n-1}^i(k)=z^i(kM-1)-\left(\boldsymbol{z}_{kM-1}^i\right)^T\boldsymbol{z}_{(k-\mu)M-2:(k-\mu)M-M_n}^i \times\left[\left(\boldsymbol{z}_{kM-2:kM-M_n}^i\right)^T\boldsymbol{z}_{(k-\mu)M-2:(k-\mu)M-M_n}^i\right]^{-1}\begin{bmatrix} z^i(kM-2) \\ \vdots \\ z^i(kM-M_n)\end{bmatrix}$$

On the other hand, clearly it follows from Eqs. 7, 11, 12 and 14 that $z^i(kM-1)=y_i(k)$,

$$\boldsymbol{\xi}_n^i(k)=\begin{bmatrix} z^i(kM-2) \\ \vdots \\ z^i(kM-M_n)\end{bmatrix},\ \boldsymbol{\rho}_n^i(k)=\begin{bmatrix} z^i((k-\mu)M-2) \\ \vdots \\ z^i((k-\mu)M-M_n)\end{bmatrix}$$

and

$$\left[\hat{\boldsymbol{\theta}}_{n,iv}^i(k)\right]^T = \left(\left\{\frac{1}{k}\sum_{t=1}^{k}\boldsymbol{\rho}_n^i(t)\left[\boldsymbol{\xi}_n^i(t)\right]^T\right\}^{-1}\frac{1}{k}\sum_{t=1}^{k}\boldsymbol{\rho}_n^i(t)y_i(t)\right)^T$$

$$= \left(z_{kM-1}^i\right)^T z_{(k-\mu)M-2:(k-\mu)M-M_n}^i$$

$$\cdot\left(\left(z_{kM-2:kM-M_n}^i\right)^T z_{(k-\mu)M-2:(k-\mu)M-M_n}^i\right)^{-1} \quad (18)$$

where we introduce a time index k to the estimated parameters $\hat{\boldsymbol{\theta}}_{n,iv}^i(k)$ in order to emphasize that the parameters are estimated from k samples of data. Hence,

$$\epsilon_{1,M_n-1}^i(k) = y_i(k) - \left(\hat{\boldsymbol{\theta}}_{n,iv}^i(k)\right)^T \boldsymbol{\xi}_n^i(k), \quad (19)$$

which is equal to $e_n^i(k)$ as given by Eqs. 6 and 12, that is, $\forall i = 1\cdots m$,

$$\epsilon_{1,M_n-1}^i(k) \equiv e_n^i(k) \quad (20)$$

Equations 18 and 20 link the ith residual $e_n^i(k)$ of the considered system and the lattice filter, making the generation of $e_n^i(k)$ possible by the use of the IV lattice filter without identifying the parameters $\boldsymbol{\theta}_n^i$ explicitly.

New notations and definitions

In addition to the forward error vector $\boldsymbol{f}_{s,\eta}^i(k)$, the complete lattice filter algorithm also includes other quantities.

Define

$$\boldsymbol{b}_{s,\eta}^i(k) = z_{kM-s-\eta}^i - z_{kM-s-\eta}^i|_{\boldsymbol{h}_{s-1,\eta}^i(k)},$$
$$\eta = 1\cdots M_n - 1;\ s = 1\cdots M$$

as the backward error vector, which is the difference between the vector $z_{kM-s-\eta}^i$ and its nonsymmetric projection onto the subspace $\boldsymbol{h}_{s-1,\eta}^i(k)$. Geometrically, $\boldsymbol{b}_{s,\eta}^i(k)$ is perpendicular to the subspace $\boldsymbol{h}_{s-1,\eta}^i(k-\mu)$.

Similarly, denote the first element of $\boldsymbol{b}_{s,\eta}^i(k)$ by

$$\gamma_{s,\eta}^i(k) = \boldsymbol{\phi}^T \boldsymbol{b}_{s,\eta}^i(k)$$

Furthermore, define

$$\boldsymbol{\phi}_{\boldsymbol{h}_{s,\eta}^i(k)} = \boldsymbol{\phi} - \boldsymbol{\phi}|_{\boldsymbol{h}_{s,\eta}^i(k)}$$

which is orthogonal to the subspace $\boldsymbol{h}_{s,\eta}^i(k-\mu)$.

Finally, we need to define several scalar quantities, which are inner products of the associated vectors,

$$\alpha_{s,\eta}^i(k) = \langle \boldsymbol{f}_{s,\eta}^i(k-\mu), \boldsymbol{f}_{s,\eta}^i(k)\rangle, \qquad s = 1\cdots M$$

$$\beta_{s,\eta}^i(k) = \langle \boldsymbol{b}_{s,\eta}^i(k-\mu), \boldsymbol{b}_{s,\eta}^i(k)\rangle, \qquad s = 1\cdots M$$

$$\overline{\omega}_{s,\eta}^{i,\mu}(k) = \begin{cases} \langle \boldsymbol{f}_{s,\eta-1}^i(k-\mu), \boldsymbol{b}_{s+1,\eta-1}^i(k)\rangle, & s = 1\cdots M-1 \\ \langle \boldsymbol{f}_{s,\eta-1}^i(k-\mu), \boldsymbol{b}_{1,\eta-1}^i(k-1)\rangle, & s = M \end{cases}$$

$$\epsilon_{s,\eta}^{i,\mu}(k) = \langle \boldsymbol{f}_{s,\eta}^i(k-\mu), \boldsymbol{\phi}_{\boldsymbol{H}_{s,\eta}^i(k)}\rangle, \qquad s = 1\cdots M$$

$$\varsigma_{s,\eta}^{i,\mu}(k) = \begin{cases} \langle \boldsymbol{b}_{s+1,\eta}^i(k-\mu), \boldsymbol{\phi}_{\boldsymbol{H}_{s,\eta}^i(k)}\rangle, & s = 1\cdots M-1 \\ \langle \boldsymbol{b}_{1,\eta}^i(k-1-\mu), \boldsymbol{\phi}_{\boldsymbol{H}_{s,\eta}^i(k)}\rangle, & s = M \end{cases}$$

and

$$\pi_{s,\eta}^i(k) = 1 - \langle \phi_{\boldsymbol{H}_{s-1,\eta}^i(k-\mu)}, \phi_{\boldsymbol{H}_{s-1,\eta}^i(k)}\rangle, \qquad s = 1\cdots M.$$

The complete set of lattice filtering algorithms for the calculation of $\boldsymbol{e}_n(k)$, including order-recursion and time-recursion equations, and the initialization, are given in the Appendices.

IV Lattice Filter-Based Process Monitoring

As discussed previously, our proposed IV lattice filter can recursively compute

$$\boldsymbol{e}_n(k) \equiv \begin{bmatrix} \epsilon_{1,M_n-1}^1(k) \\ \vdots \\ \epsilon_{1,M_n-1}^m(k) \end{bmatrix} \equiv \boldsymbol{\epsilon}_{1,M_n-1}(k)$$

This residual error signal is now used to construct a statistic for process monitoring.

Monitoring index

As shown by the last line of Eq. 8, under the normal case $\boldsymbol{e}_n(k)$ is an MA process with simultaneous excitation by noise vectors $\boldsymbol{o}(k)$, $\boldsymbol{v}(k)$, and $\boldsymbol{p}(k)$. If any of these noise vectors is a multivariate zero-mean Gaussian process, it must follow that

$$\boldsymbol{e}_n(k) \sim \aleph\left[\boldsymbol{0}, \boldsymbol{R}_n^e(k)\right]$$

and consequently

$$\boldsymbol{e}_n^T(k)\left[\boldsymbol{R}_n^e(k)\right]^{-1}\boldsymbol{e}_n(k) \sim \chi^2(m)$$

However, the true value of $\boldsymbol{R}_n^e(k)$ is usually unknown. To work around this issue, we use the Hotelling T^2 statistic as the index for process monitoring, which is based on the estimated covariance matrix. Moreover, note that $\boldsymbol{e}_n(k)$ being an nth-order MA process, it is autocorrelated with $\boldsymbol{e}_n(k+\tau)$ for any $0 \le \tau \le n$. Therefore, at the kth time instant, the following residual vectors sequence

$$\{\boldsymbol{e}_n(0)\ \boldsymbol{e}_n(1)\cdots \boldsymbol{e}_n(k-1)\ \boldsymbol{e}_n(k)\}$$

cannot be directly used to compute the T^2 statistic (Johnson and Wichern, 1998). To work with a series of uncorrelated signals from the aforementioned sequence, we consider a series of lagged signals, that is,

$$\left\{\boldsymbol{e}_n(0)\ \boldsymbol{e}_n(\tau)\cdots \boldsymbol{e}_n\left[\mathrm{Int}\left(\frac{k-\tau}{\tau}\right)\tau\right]\ \boldsymbol{e}_n\left[\mathrm{Int}\left(\frac{k}{\tau}\right)\tau\right]\right\}$$

for the computation of the T^2 statistic. Clearly the disadvantage of this approach is the time delay in detecting a fault during monitoring. For instance, if the process has an abnormal variation at time instant k, then such a variation will not be detected until the time instant $k+\tau$. Generally, since τ is selected to be $n+1$, if the process order n is not high, a time delay of this magnitude can be tolerated.

At time instant k_0, assume that a sequence of uncorrelated residual vectors,

$$\left\{ \boldsymbol{e}_n(0)\ \boldsymbol{e}_n(\tau) \cdots \boldsymbol{e}_n\left[\mathrm{Int}\left(\frac{k_0}{\tau}\right)\tau - \tau\right]\ \boldsymbol{e}_n\left[\mathrm{Int}\left(\frac{k_0}{\tau}\right)\tau\right]\right\}$$

has been computed when a process is normal. Now the goal is to use this residual sequence to determine a nominal operation or control region as a template for comparing with future residuals

$$\left\{ \boldsymbol{e}_n\left(\mathrm{Int}\left(\frac{k_0}{\tau}\right)\tau + \tau\right)\ \boldsymbol{e}_n\left(\mathrm{Int}\left(\frac{k_0}{\tau}\right)\tau + 2\tau\right) \cdots \boldsymbol{e}_n\left(\mathrm{Int}\left(\frac{k}{\tau}\right)\tau\right)\right\}$$

We have to emphasize that "raw" residual monitoring alone can lead to erroneous results. Thus simple residual monitoring is not enough, because each variable can play a different role in the process. For instance, while a small-sized residual in one variable means a significant deviation in the variable from its desired value, a large-sized residual in another variable may only indicate a slight deviation of the variable from its target. To avoid this conflict, one solution, in monitoring the process, is to scale all the variables to have an identical weight, as in the Hotelling's $T_n^2(k)$ shown later.

We denote $\kappa_1 \equiv \mathrm{Int}(k_0/\tau)\tau$ and we select $\mathrm{Int}(k_0/\tau)$ uncorrelated residual vectors to compute

$$\bar{\boldsymbol{e}}_n(\kappa_1) = \frac{1}{\mathrm{Int}\left(\frac{k_0}{\tau}\right)} \sum_{i=0}^{\mathrm{Int}(k_0/\tau)-1} \boldsymbol{e}_n\left\{\kappa_1 + \left[i - \mathrm{Int}\left(\frac{k_0}{\tau}\right) + 1\right]\tau\right\} \tag{21}$$

with $\tau = n+1$.

According to Johnson and Wichern (1998)

$$T_n^2(k) \equiv \frac{\mathrm{Int}\left(\frac{k_0}{\tau}\right)}{\mathrm{Int}\left(\frac{k_0}{\tau}\right)+1}\left[\boldsymbol{e}_n(k) - \bar{\boldsymbol{e}}_n(\kappa_1)\right]^T\left[\hat{\boldsymbol{R}}_n^e(\kappa_1)\right]^{-1} \times \left[\boldsymbol{e}_n(k) - \bar{\boldsymbol{e}}_n(\kappa_1)\right] \tag{22}$$

is distributed as

$$\frac{\mathrm{Int}\left(\frac{k_0}{\tau}\right) - 1m}{\mathrm{Int}\left(\frac{k_0}{\tau}\right) - m} F_{m,\mathrm{Int}}\ (k_0/\tau) - m,$$

where (1) $F_{m,\mathrm{Int}(k_0/\tau)-m}$ stands for an F-distributed random variable with degrees of freedom m and $\mathrm{Int}(k_0/\tau) - m$; and (2)

$$\hat{\boldsymbol{R}}_n^e(\kappa_1) = \frac{1}{\mathrm{Int}\left(\frac{k_0}{\tau}\right) - 1} \sum_{i=0}^{\mathrm{Int}(k_0/\tau)-1} \times \left\{\boldsymbol{e}_n\left[\kappa_1 + \left(i - \mathrm{Int}\left(\frac{k_0}{\tau}\right) + 1\right)\tau\right] - \bar{\boldsymbol{e}}_n(\kappa_1)\right\} \times \left\{\boldsymbol{e}_n\left[\kappa_1 + \left(i - \mathrm{Int}\left(\frac{k_0}{\tau}\right) + 1\right)\tau\right] - \bar{\boldsymbol{e}}_n(\kappa_1)\right\}^T \tag{23}$$

is the estimate of covariance matrix of $\boldsymbol{e}_n(t)$ based on a sequence of $\mathrm{Int}(k_0/\tau)$ uncorrelated residual vectors.

Therefore, given a level of significance δ, the confidence limit of $T_n^2(k)$ for process monitoring will be

$$\frac{\left(\mathrm{Int}\left(\frac{k_0}{\tau}\right) - 1\right)m}{\mathrm{Int}\left(\frac{k_0}{\tau}\right) - m} F_{m,\mathrm{Int}(k_0/\tau)-m}(\delta),$$

which is a function of $\mathrm{Int}(k_0/\tau)$ and m, the dimension of the residual vector $\boldsymbol{e}_n(\cdot)$.

As each newly computed residual vector becomes available, we can compute $T_n^2(k)$ and check if it stays within the control region. In this way, the process under consideration can be effectively monitored.

Treatment of some practical issues

When we apply the proposed approach to monitoring a practical process, some practical issues need to be considered.

Preprocessing of Process Data. We denote the process variables sampled at the tth instant by $z(t)$, which includes both inputs and outputs. First of all, outliers in the data should be removed. Many data sets may contain unusual observations that do not seem to belong to the pattern of variability produced by the other observations. These unusual observations are referred to as *outliers*. For data with a single characteristic, outliers are those that are either very large or very small relative to the others. The situation can be more complicated with multivariate data. Outliers are best detected visually whenever this is possible. When the number of observations k is large, scatter plots are not feasible. When the number of variables $m+l$ (number of inputs and outputs) is large, the large number of scatter plots $(m+l)(m+l-1)/2$ may prevent viewing them all. Even so, we suggest visual inspection of the data whenever possible. If the process variables are high-dimensional, outliers cannot be detected from the univariate or bivariate scatter plots easily. However, in this case, assume that we have k samples of data. Then a large value of $\{[z(t) - \bar{z}]^T \boldsymbol{S}^{-1}[z(t) - \bar{z}]\}$ for $t \in [0,k]$ will suggest an unusual observation (Johnson and Wichern, 1998), even though it cannot be seen visually, where $\bar{z}$ and $\boldsymbol{S}$ are the mean and covariance matrix of z estimated over $t \in [0,k]$.

Second, some process data may be missing. Since correlations exist among multivariate data of process variables, we can make use of these correlations to reconstruct the missing data. For example, the approach proposed by Dunia et al. (1996) and Burnham et al. (1999) can be used for data reconstruction.

Finally, we need to ensure that each output variable is linearly independent of the others. For real data, if some elements of $\boldsymbol{y}(k)$ are correlated, data preprocessing should be carried out. For example, we can perform PCA on the sampled $\boldsymbol{y}(k)$ to remove its correlated elements. Consequently, instead of $\boldsymbol{y}(k)$, the scores $\hat{\boldsymbol{p}}_y^T\boldsymbol{y}(k)$ will be fed to the lattice filters, where $\hat{\boldsymbol{p}}_y$ are the principal eigenvectors of the covariance matrix of $\boldsymbol{y}(k)$. Similarly, we can preprocess the sampled input $\boldsymbol{u}(k)$ if it has correlated elements so that not $\boldsymbol{u}(k)$ but its scores $\hat{\boldsymbol{p}}_u^T\boldsymbol{u}(k)$ will be fed to the lattice filters, where $\hat{\boldsymbol{p}}_u$ are the principal eigenvectors of the covariance matrix of $\boldsymbol{u}(k)$. Alternately, data prefiltering can also be used to remove the correlation.

Online Recursive Order Determination for Lattice Filters. When applying the proposed lattice filter-based approach to the monitoring of a process, since the process order n is usually unknown *a priori* and can vary, we have to estimate it in real time and on line.

The correct choice of a process order is very important. If the chosen order is too low, the process will be underparametrized, and as a consequence, the unmodeled dynamics of the process may destabilize. This means that in practice there maybe a tendency to overparametrize the process. However, with overparametrization, there is a danger of ill-conditioning in the lattice filter algorithm, and the convergence of the algorithm cannot be guaranteed (Xia and Moore, 1989).

Chen and Guo (1987) have proposed an approach based on the Bayesian information criterion (BIC) for a consistent estimation of the order of a class of stochastic systems given a batch of data. We adopt this approach here for the on line recursive determination of the order of the lattice filter.

From $\boldsymbol{e}_n(k)\equiv\boldsymbol{\epsilon}_{1,M_n-1}(k)$, it turns out that the process order n corresponds to the order index $\eta=M_n-1$ of the lattice filter. Therefore, the determination of n is in a one-to-one correspondence with the determination of η.

With an available sequence of residual vectors $\{\boldsymbol{\epsilon}_{1,M_n-1}(0)\cdots\boldsymbol{\epsilon}_{1,M_n-1}(k)\}$, we define a function

$$\Omega_k(\eta)=k\log\left[\sum_{t=0}^{k}\boldsymbol{\epsilon}_{1,\eta}^T(t)\boldsymbol{\epsilon}_{1,\eta}(t)\right]+2\left(\frac{\eta+1}{M}-1\right)\log(k)\log[\log(k)] \quad (24)$$

Then the lattice filter order η_0 will be determined by minimizing $\Omega_k(\eta)$ with respect to $\eta\in[M-1,\ \dots\ ,M_L-1]$, for example,

$$\eta_0=\arg\min(\Omega_k(\eta)),\ \eta\in[M-1,\ \dots\ ,M_L-1] \quad (25)$$

where L can be a large positive integer and $M_L=M(L+1)$. Moreover, since the process order n can vary, we need to develop a recursive scheme for order determination as new data become available.

Intuitively, for a time-varying process, its order change is less frequent than its parameter change. Therefore, we propose a batchwise approach to recursive-order determination. Define an intermediate variable

$$\Xi\boldsymbol{\epsilon}(k)\equiv\sum_{t=0}^{k}\boldsymbol{\epsilon}_{1,\eta}^T(t)\boldsymbol{\epsilon}_{1,\eta}(t) \quad (26)$$

We can rewrite Eq. 24 into

$$\Omega_k(\eta)=k\log\Xi\boldsymbol{\epsilon}(k)+2\left(\frac{\eta+1}{M}-1\right)\log(k)\log(\log(k)) \quad (27)$$

If at time instant $k+n_{k+1}$, a new sequence of residual vectors $\{\boldsymbol{\epsilon}_{1,M_n-1}(k+1)\cdots\boldsymbol{\epsilon}_{1,M_n-1}(k+n_{k+1})\}$ becomes available $(n_{k+1}\geq 1)$, in terms of Eq. 26, we have

$$\Xi\boldsymbol{\epsilon}(k+n_{k+1})\equiv\Xi\boldsymbol{\epsilon}(k)+\sum_{t=k+1}^{k+n_{k+1}}\boldsymbol{\epsilon}_{1,\eta}^T(t)\boldsymbol{\epsilon}_{1,\eta}(t) \quad (28)$$

Therefore,

$$\Omega_{k+\lambda_{k+1}}(\eta)=(k+n_{k+1})\log\left[\Xi\boldsymbol{\epsilon}(k)+\sum_{t=k+1}^{k+n_{k+1}}\boldsymbol{\epsilon}_{1,\eta}^T(t)\boldsymbol{\epsilon}_{1,\eta}(t)\right]+2\left(\frac{\eta+1}{M}-1\right)\log(k+n_{k+1})\log[\log(k+n_{k+1})] \quad (29)$$

Using Eq. 28, $\Omega_{k+n_{k+1}}(\eta)$ is recursively computed. Eventually, if the η_0 corresponding to the minimum of $\Omega_{k+n_{k+1}}(\eta)$ is found, as shown in Eq. 25, then the new process order will be updated to

$$n_0=\frac{\eta_0+1}{M}-1$$

Update of Covariance Matrix. Since the process under surveillance is time-varying, we also need to recursively update the covariance matrix $\hat{\boldsymbol{R}}_n^e(\mathrm{Int}(k_0/\tau)\tau)$, based on the newly sampled process data. This will provide the proposed monitoring scheme an adaptive capability. As the time instant k progresses, some old data should be discounted and the most recent data should be *weighted* heavily in the recursive updates. When we update the covariance matrix using a sequence of w_l uncorrelated residual vectors at one time, at the time instant $\kappa_2=\kappa_1+w_l\tau$, we have

$$\hat{\boldsymbol{R}}_n^e(\kappa_2)=\frac{\frac{\kappa_1}{\tau}-1}{\frac{\kappa_2}{\tau}-1}\hat{\boldsymbol{R}}_n^e(\kappa_1)+\frac{\frac{\kappa_1}{\tau}}{\frac{\kappa_2}{\tau}-1}\left[\bar{e}_n(\kappa_1)-\bar{e}_n^T(\kappa_2)\right]\times\left[\bar{e}_n(\kappa_1)-\bar{e}_n^T(\kappa_2)\right]^T+\frac{1}{\frac{\kappa_2}{\tau}-1}\sum_{i=(\kappa_2/\tau)-w_l}^{(\kappa_2/\tau)-1}\times\left\{\boldsymbol{e}_n\left[\kappa_1+\left(i-\frac{\kappa_2}{\tau}+1+w_l\right)\tau\right]-\bar{e}_n(\kappa_1)\right\}\times\left\{\boldsymbol{e}_n\left[\kappa_1+\left(i-\frac{\kappa_2}{\tau}+1+w_l\right)\tau\right]-\bar{e}_n(\kappa_1)\right\}^T \quad (30)$$

and the mean update equation is

$$\bar{e}_n(\kappa_2)=\frac{\kappa_1}{\kappa_2}\bar{e}_n(\kappa_1)+\frac{\tau}{\kappa_2}\sum_{i=(\kappa_2/\tau)-w_l}^{(\kappa_2/\tau)-1}\times e_n\left[\kappa_1+\left(i-\frac{\kappa_2}{\tau}+1+w_l\right)\tau\right] \quad (31)$$

The derivation of Eqs. 30 and 31 is straightforward, and is not included here due to lack of space.

Adaptive process monitoring

Once the data preprocessing is completed, the sequence of steps required for real-time and on line adaptive process monitoring is as follows.

1. Choose an initial data block with d_0 samples and apply the proposed lattice filtering algorithm given in the Appendices to the first segment of the block to generate residual vectors. Determine the order of the lattice filter from the generated residual vectors using the index given in Eq. 27.
2. With the determined n, let τ be $n+1$. Apply the lattice filter to the remaining data in the initial data block. Select $\mathrm{Int}(d_0/\tau)$ uncorrelated residual vectors to estimate the covariance matrix $\boldsymbol{R}_n^e(\mathrm{Int}(d_0/\tau)\tau)$ according to Eq. 23.
3. As new data samples become available, calculate the residual vectors based on the lattice filters. Select w_l, and calculate $T_n^2(k)$ and compare it with its predetermined confidence limit. If $T_n^2(k)$ is within its confidence limit, recursively update the covariance matrix with w_l uncorrelated residual vectors and the order with n_{k+1} residual vectors, according to Eqs. 30 and 29, respectively. Otherwise, stop updating, announce alarm, and take a further action for diagnosis if necessary.
4. The aforementioned monitoring procedure is repeated for each new data sample.

Case Studies

In this section, we conduct two case studies to illustrate the application of the proposed IV lattice filter to dynamic process monitoring. The objective of the studies is to show the lattice filter's ability to adapt to slow, normal process changes and detect abnormal operations. The first case study is based on a simulated process, and the second one is based on a real pilot plant.

Case study one

A second-order dynamic process with four inputs and four outputs, for example

$$x(k+1)=\begin{bmatrix}0.67 & 0.67\\ -0.67 & 0.67\end{bmatrix}x(k)+\begin{bmatrix}-0.4326 & 0.1253 & -1.1456 & 1.1892\\ -1.6656 & 0.2877 & 1.1909 & -0.0376\end{bmatrix}\tilde{u}(k)+p(k)$$

$$\tilde{y}(k)=\begin{bmatrix}0.3573 & -0.5883\\ 0.1746 & 2.1832\\ -0.1867 & -0.1364\\ 0.7258 & 0.1139\end{bmatrix}x(k)+\begin{bmatrix}1.0668 & 0.2944 & -0.6918 & -1.4410\\ 0.0593 & -1.3362 & 0.8580 & 0.5711\\ -0.0956 & 0.7143 & 1.2540 & -0.3999\\ -0.8323 & 1.6236 & -1.5937 & 0.6900\end{bmatrix}\tilde{u}(k) \quad (32)$$

is used to generate data. Further, according to Eq. 3-4, we convert Eq. 32 into the following input–output equation:

$$\begin{aligned}\tilde{y}(k)-1.34\tilde{y}(k-1)+0.8978\tilde{y}(k-2)&=\begin{bmatrix}1.0668 & 0.2944 & -0.6918 & -1.4410\\ 0.0593 & -1.3362 & 0.8580 & 0.5711\\ -0.0956 & 0.7143 & 1.2540 & -0.3999\\ -0.8323 & 1.6236 & -1.5937 & 0.6900\end{bmatrix}\tilde{u}(k)\\ &+\begin{bmatrix}-0.5912 & -0.5227 & -0.1485 & 2.3423\\ -3.7913 & 2.4405 & 1.2502 & -0.6397\\ 0.4361 & -1.0198 & -1.6289 & 0.3190\\ 0.6116 & -2.0519 & 1.4397 & -0.0658\end{bmatrix}\tilde{u}(k-1)\\ &+\begin{bmatrix}-0.1396 & 0.4627 & -0.0909 & -1.1088\\ 2.9781 & -1.7848 & 0.9774 & -1.3153\\ -0.1233 & 0.6587 & 0.8377 & -0.1003\\ -1.1867 & 1.5051 & -0.2981 & -0.0650\end{bmatrix}\tilde{u}(k-2)\\ &+\begin{bmatrix}0.3273 & -0.5883 & 0.1749 & 0.6135\\ 0.1746 & 2.1832 & -1.5797 & -1.3458\\ -0.1867 & -0.1364 & 0.2165 & -0.0337\\ 0.7258 & 0.1139 & -0.5626 & 0.4100\end{bmatrix}\begin{bmatrix}p(k-1)\\ p(k-2)\end{bmatrix}\end{aligned} \quad (33)$$

We select a combination of sine waves with different frequencies as the noise-free input sequence $\{\tilde{u}(k)\}$. We assume that the observed inputs $u(k)$ and outputs $y(k)$ are corrupted by noise vectors $o(k)$ and $v(k)$, respectively, for example,

$$\begin{aligned}u(k)&=\tilde{u}(k)+v(k)\\ y(k)&=\tilde{y}(k)+o(k)\end{aligned} \quad (34)$$

where $o(k)$, $v(k)$, and $p(k)$ are zero-mean Gaussian-distributed random vectors whose means and variances are as follows:

$$o(k)\sim\aleph\left(\mathbf{0},\begin{bmatrix}0.15^2 & 0 & 0 & 0\\ 0 & 0.4^2 & 0 & 0\\ 0 & 0 & 0.08^2 & 0\\ 0 & 0 & 0 & 0.15^2\end{bmatrix}\right)$$

$$v(k)\sim\aleph\left(\mathbf{0},\begin{bmatrix}0.04^2 & 0 & 0 & 0\\ 0 & 0.04^2 & 0 & 0\\ 0 & 0 & 0.04^2 & 0\\ 0 & 0 & 0 & 0.04^2\end{bmatrix}\right)$$

$$p(k)\sim\aleph\left(\mathbf{0},\begin{bmatrix}0.15^2 & 0\\ 0 & 0.17^2\end{bmatrix}\right)$$

We select an initial block containing the first 300 input and output points as training data. We apply the IV lattice filter to this block for the generation of the residual vector $\boldsymbol{\epsilon}_{1,\eta}(k)$. Since the process has four inputs and four outputs, $M=8$ and $\eta=1\cdots 8(n+1)-1$.

We determine the initial process order n, which is two, using the first 100 data points in the block. Then, we estimate the initial value of the covariance matrix $\boldsymbol{R}_n^e(\text{Int}(300/\tau)\tau)$ with $\tau = 3$ using $\text{Int}(300/\tau) = 100$ uncorrelated residual vectors.

We consider two scenarios of parameter variation. First, starting at the time instant $k = 1{,}501$ until $k = 6{,}000$, we continuously change the parameters of the following system matrix

$$\boldsymbol{A} = \begin{bmatrix} 0.67 & 0.67 \\ -0.67 & 0.67 \end{bmatrix}$$

at each time instant. For example, at the kth time instant and thereafter at each time step, the first, second, and fourth elements decrease by 0.01%, but the third element increases by 0.01% of their respective values at the $(k-1)$-th time instant. Consequently, at the time instant $k = 6{,}000$, we have

$$\boldsymbol{A} = \begin{bmatrix} 0.4272 & 0.4272 \\ -0.4272 & 0.4272 \end{bmatrix}$$

We consider such a parametric variation as a normal process drift because its magnitude is tiny.

Second, at the $k = 8{,}001$ time instant (no parametric changes are made during the period of time from $k = 6{,}000$ until $k = 8{,}000$.), we change the order of the system from two to three, and at the same time introduce additional parameters and process disturbance to the system. For instance, after such a change, matrices $\boldsymbol{A}$, $\boldsymbol{C}$, and the process disturbance $\boldsymbol{p}(k)$, respectively, are as follows:

$$\boldsymbol{A} = \begin{bmatrix} 0.4272 & 0.4272 & -a \\ -0.4272 & 0.4272 & a \\ a & -a & a \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} -0.4326 & 0.1253 & -1.1456 & 1.1892 \\ -1.6656 & 0.2877 & 1.1909 & -0.0376 \\ b_1 & b_2 & b_3 & b_4 \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} 0.3273 & -0.5883 & 0.2225 \\ 0.1746 & 2.1832 & -0.4532 \\ -0.1867 & -0.1364 & 0.2132 \\ 0.7258 & 0.1139 & 0.5674 \end{bmatrix}$$

$$\boldsymbol{p}(k) \backsim \aleph\left(\boldsymbol{0}, \begin{bmatrix} 0.15^2 & 0 & 0 \\ 0 & 0.17^2 & 0 \\ 0 & 0 & 0.08^2 \end{bmatrix}\right)$$

where $a = 0.44$, $b_1 = -1.2332$, $b_2 = 0.4532$, $b_3 = 0.0987$, and $b_4 = -0.9987$.

We use two approaches to monitor the previously mentioned time- and order-varying process simultaneously, and the associated results are shown in Figures 1 and 2, respectively. Note that in both figures, (1) 95% and 99% confidence limits for $T_n^2(k)$ are selected; (2) the covariance matrix is updated based on $w_l = 100$ uncorrelated residual vectors if no alarm is announced; and (3) after every $n_{k+1} = 100$ samples, an on-line order redetermination is carried out.

In Figure 1, the residual vector $\boldsymbol{e}_n(k)$ is generated according to Eq. 5 based on the known state-space model matrices $\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}\}$. Since the model is constant, the calculated $T_n^2(k)$ frequently exceeds its confidence limit after the time instant $k = 1{,}500$, even though the process is operating normally, giving a lot of false alarms. Such false alarms would not be tolerated in the industry.

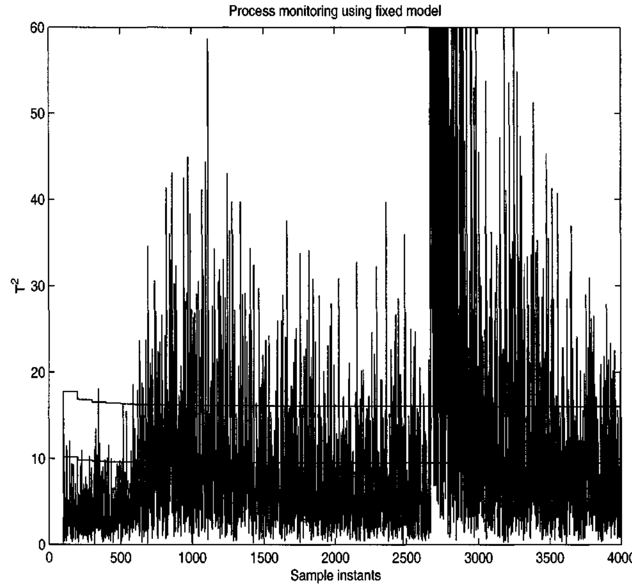

Figure 1. Constant model-based process monitoring for simulated system.

Figure 2 shows the $T_n^2(k)$ calculated from the residual vector generated by the adaptive IV lattice filter with a forgetting factor $\lambda = 0.99$. Since the IV lattice filter can adapt to the process variation, the T_n^2 statistic is basically within its 99% confidence limit for $k \leq 6{,}000$, significantly eliminating false alarms. Further, when the order of the system changes at the time instant $k = 8{,}001$, we determine the new order at the time instant $k = 8{,}100$ using the index given by Eq. 29. As shown clearly in the figure, the IV lattice filter-based $T_n^2(k)$ triggers its control limit during the transient period of time and drops back to the limit after the system enters its new

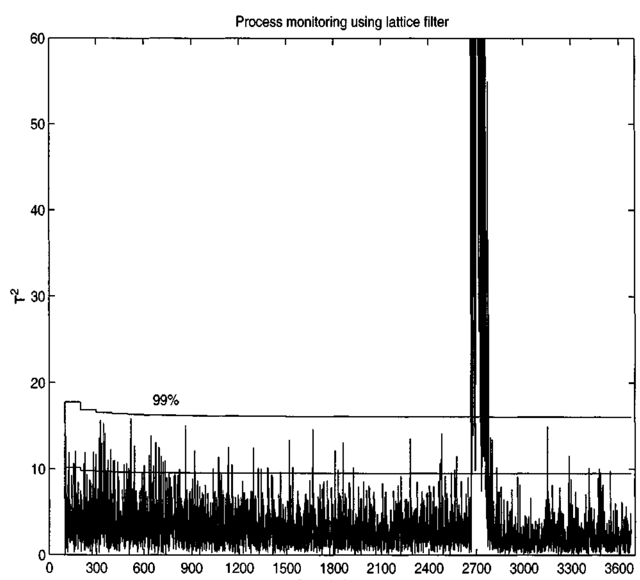

Figure 2. Adaptive IV lattice filter-based process monitoring for simulated system.

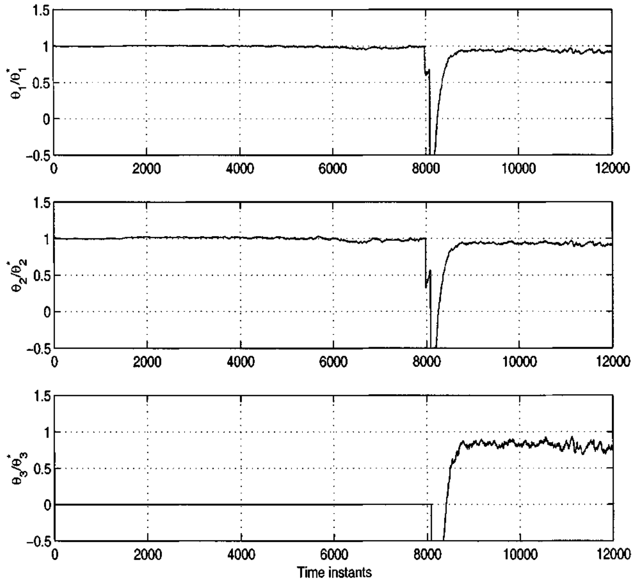

Figure 3. Tracking parameters of the AR part of simulated system.

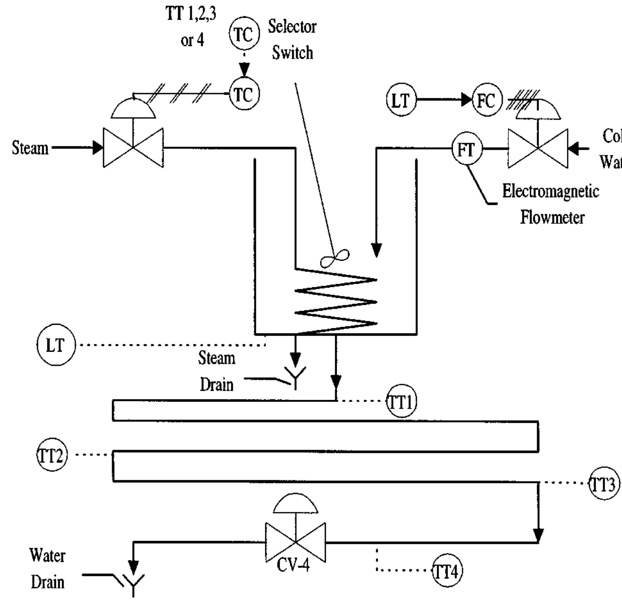

Figure 4. CSTHS.

steady state, tracking the operational state of the system effectively. Note that in the x-axis of Figures 1 and 2, each point is not equal to each sample of the original data. Since we compute $T_n^2(k)$ using a sequence of uncorrelated residual vectors, before (including) point 2,700, each point in the x-axis corresponds to three data samples ($\tau = 3$), and after point 2,700, each point corresponds to four data samples ($\tau = 4$).

We also perform an experiment to observe how the lattice filter tracks the parameters of the AR part in the ARMAX process. It is seen from Eq. 33 that the initial values of the AR parameters are $\alpha_1 = -1.34$ and $\alpha_2 = 0.8978$, respectively. When we change the parameters of the A matrix, the two parameters are also subject to a slight variation, for example, at the instant $k = 6{,}000$, $\alpha_1 = -0.8544$, and $\alpha_2 = 0.3650$. Moreover, at the instant $k = 8{,}001$, when the order of the simulated process is changed from 2 to 3 suddenly, besides an extra parameter $\alpha_3 = -0.326$ being introduced to the AR part, α_1 and α_2 become -1.294 and 0.3650, respectively. Figure 3 plots the ratios of the estimated α_1, α_2, and α_3 over their respective true values, from which we find that the slowly time-varying parameters α_1, α_2, and α_3 are effectively tracked.

Case study two

In this study, we use the proposed scheme to monitor a real continuous stirred-tank heater system (CSTHS). The system is located in the Computer Process Control Laboratory, in the Department of Chemical and Materials Engineering, University of Alberta, Canada, and is shown in Figure 4, where the cold water continuously flowing through the tank is heated by high-temperature steam passing through a coil and four thermocouples (e.g., TT1, TT2, TT3, and TT4 in Figure 4) installed at different locations of the long exit pipe provide temperature signals. The ultimate purpose of the CSTHS is to control the level and temperature of the water in the tank. There are three PID controllers included in the system. Two of them control the level and the temperature of the water in the tank, respectively, and the third one controls the cold water flow rate.

In Figure 5, the lower subplot illustrates the flow rates of steam and cold water. As mentioned above, these are the manipulated variables for controlling temperature and level of the water in the tank. The upper subplot shows the corresponding level and temperature of the cold water.

The purpose of this case study is twofold. First, to show again that our proposed adaptive monitoring scheme will not cause any false alarms when applied to a real pilot plant with a normal drift. Then, we show that in the event of a real fault, the proposed scheme can immediately detect the fault.

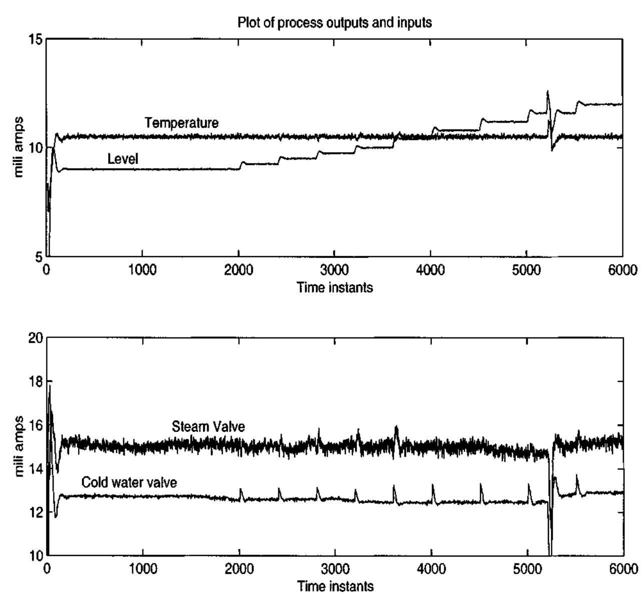

Figure 5. Flow rates of steam and cold water; level and temperature of cold water in CSTHS.

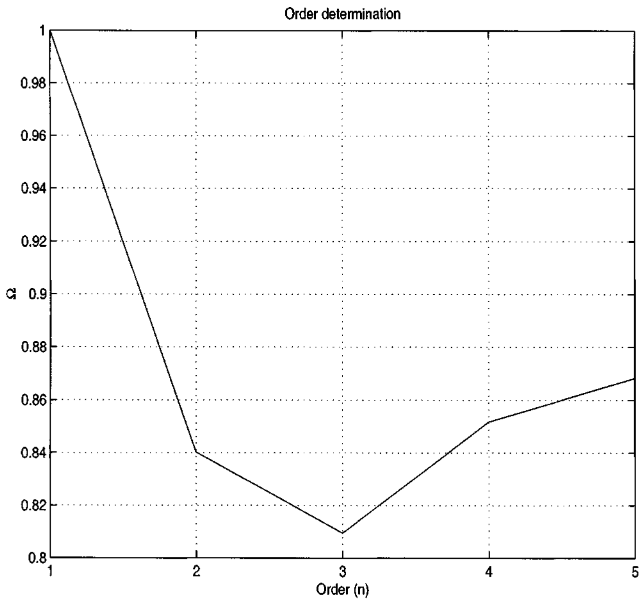

Figure 6. Order determination of CSTHS model.

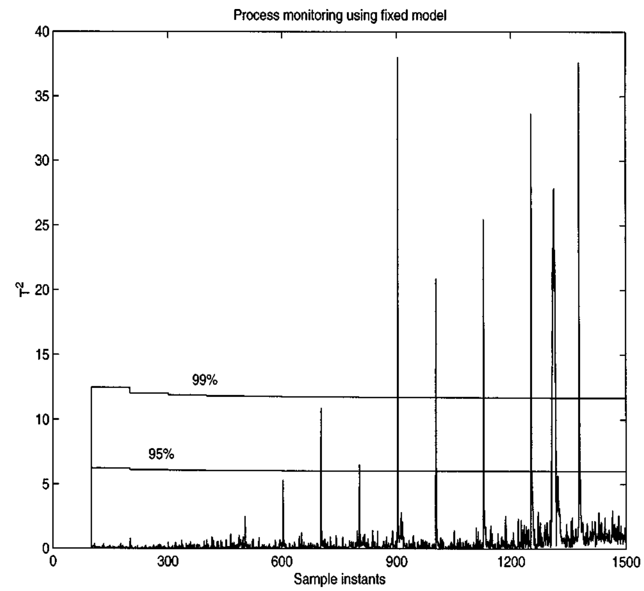

Figure 7. Constant model-based process monitoring for CSTHS (using every fourth sample period).

We collect 300 samples from the CSTHS as the initial data block. Since the lattice filter converges very quickly, we only use the first 100 samples in the block to identify the order and coefficients of the lattice filter. The result of order determination is illustrated in Figure 6, where apparently the order is selected to be three based on our previous analysis. Further, we estimate the initial value of the covariance matrix of the residual vectors based on the whole block of data.

After time instant $k = 1{,}500$, we introduce a slight drift into the system by adding a small ramp to the level output, which is fed back to the level PID controller. Then, after time instant $k = 2{,}000$, we change the setpoint of the water level several times by a magnitude 0.25 each time. Further, we change the setpoint with a magnitude 0.40 each time after the time instant 3,600. These setpoint changes in the level result in significant overall changes to the process. For example, when the level changes each time, the temperature-to-steam loop gain and time constant also change, making the CSTHS a time-varying system. However, we require that these setpoint changes and a slight process drift should be considered normal and not process faults from an engineering point of view.

As in the previous case study, we use two schemes to monitor the CSTHS simultaneously, where again (1) uncorrelated residual vectors generated from the lattice filter are used to compute $T_n^2(k)$; (2) $w_l = 100$ uncorrelated residual vectors are used to update the covariance matrix; and (3) $n_{k+1} = 100$ residual vectors are employed to redetermine on-line the order at one time. In the first scheme, the lattice filter identified from the 100 samples in the initial block is used as a fixed model to generate the residual vector, and the corresponding result is given in Figure 7. Since the residuals are produced from the constant model, when the CSTHS is subjected to a normal variation, the computed $T_n^2(k)$ exceeds its confidence limit, especially its 95% confidence limit, frequently. These limit violations would clearly sound many false alarms.

Figure 8 depicts the result of IV lattice filter-based process monitoring and fault detection, where as compared with Figure 7, the number of false alarms are greatly reduced. Eventually, in Figure 8, we also demonstrate how the IV lattice filter-based scheme detects fault. Between $k = 5{,}200$ and $k = 5{,}260$, a real fault is introduced into the CSTHS by choking the cold water exit pipe, and this fault is instantly detected by the statistic $T_n^2(k)$.

Conclusions

We have proposed a IV multichannel lattice filter scheme for adaptive dynamic process monitoring with two main con-

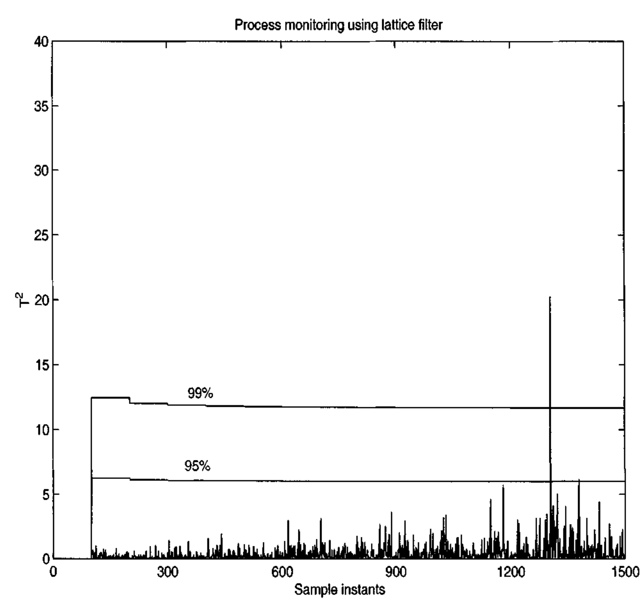

Figure 8. IV lattice filter-based monitoring for CSTHS (using every fourth sample period).

tributions. First, we develop a IV-based multichannel lattice filter that gives a consistent estimate of the vector ARMAX model in the presence of measurement noise in inputs and outputs and process noise. Then, based on the newly developed IV lattice filter, we propose an adaptive process-monitoring scheme for fully dynamic and time-varying processes.

We have successfully applied the proposed scheme to two cases, a simulated process and a real pilot plant. In comparison with a constant model-based process-monitoring technique, our proposed approach can adapt to a normal process drift, with significant reduction in the number of false alarms, and yet at the same time is sensitive to any real faults.

Currently, the IV lattice filter is updated at each data sample. In a practical situation, particularly for slowly time-varying chemical processes, we may not need to update the process models at each data point. Instead it is suggested that one update the model based on nonoverlapping data blocks of user-specified window lengths. Work on such a block-based lattice filter for adaptive processing is currently in progress.

Acknowledgment

Financial support from the Natural Science and Engineering Research Council (NSERC) of Canada is highly acknowledged.

Literature Cited

Burnham, A., J. MacGregor, and R. Viveros, "Latent Variable Multivariate Regression Modeling," *Chemometrics Intell. Lab. Syst.*, **48**, 167 (1999).

Chen, H., and L. Guo, "Consistent Estimation of the Order of Stochastic Control Systems," *IEEE Trans. Automat. Control*, ***AC*-32**, 531 (1987).

Chou, C. T., and M. Verhaegen, "Subspace Algorithms for the Identification of Multivariable Dynamic Errors-in-Variables Models," *Automatica*, **33**, 1857 (1997).

Dunia, R., J. Qin, T. Edgar, and T. McAvoy, "Identification of Faulty Sensors Using Principal Component Analysis," *AIChE J.*, **42**, 2797 (1996).

Friedlander, B., "Lattice Methods for Adaptive Processing," *Proc. IEEE*, **70**, 829 (1982).

Friedlander, B., "Lattice Implementation of Some Recursive Parameter-Estimation Algorithms," *Int. J. Control*, **37**, 661 (1983).

Gustafsson, T., M. Lovera, and M. Verhaegen, "Recursive Subspace Identification of Linear and Non-Linear Wiener State-Sapce Methods," *Automatica*, **36**, 1639 (2000).

Honig, M., and D. Messerschmitt, "Convergence Properties of an Adaptive Digital Lattice Filters," *IEEE Trans. Acoust. Speech, Signal Process.*, **ASSP-29**, 642 (1981).

Itakura, F., and S. Saito, "Digital Filtering Techniques for Speech Analysis and Synthesis," *Proc. Int. Conf. Acoustics*, Budapest, Hungary, p. 261 (1971).

Jabbari, F., and J. Gibson, "Vector-Channel Lattice Filters and Identification of Flexible Structures," *IEEE Trans. Automat. Control*, **AC-33**, 448 (1988).

Johnson, R., and D. Wichern, *Applied Multivariate Statistical Analysis*, 4th ed., Prentice Hall, Upper Saddle River, NJ (1998).

Kailath, T., *Linear Systems*, Prentice Hall, Englewood Cliffs, NJ (1980).

Kourti, T., and J. MacGregor, "Process Analysis, Monitoring and Diagnosis, Using Multivariate Projection Methods," *Chemometrics Intell. Lab. Syst.*, **28**, 3 (1995).

Kresta, J., J. MacGregor, and T. Marlin, "Multivariate Statistical Monitoring of Processes," *Can. J. Chem. Eng.*, **69**, 35 (1991).

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

Kummert, A., S. Hartwig, and A. Hypki, "A New Adaptive Lattice Filter for the ARMA Modeling of Vector Signals," *Signal Process.*, **26**, 61 (1992).

Larimore, W., "System Identification Reduced-Order Filtering and Modeling via Canonical Variate Analysis," *Proc. Amer. Control Conf.*, S. Rao and T. Torato, eds., IEEE, New York, p. 445 (1983).

Larimore, W., "Process Monitoring Using Canonical Variate Analysis and Principal Component Analysis," *Proc. 1997 IFAC ADCHEM*, Banff, Alberta, Canada (1997).

Lee, D., M. Morf, and B. Friedlander, "Recursive Least Squares Ladder Estimation Algorithms," *IEEE Trans. Acoust. Speech, Signal Process.*, **ASSP-29**, 627 (1981).

Lee, D., M. Morf, and B. Friedlander, "Recursive Ladder Estimation Algorithms for Arma Modeling," *IEEE Trans. Automat. Control*, **AC-27**, 753, (1982).

Lev-Ari, H., T. Kailath, and J. Cioffi, "Least-Squares Adaptive Lattice and Transversal Filters; A Unified Geometric Theory," *IEEE Trans. Inform. Theory*, **IT-30**, 222 (1984).

Li, W., and S. Shah, "Instrumental Variable Lattice Filter for the Identification of EIV Multivariable ARMAX Model," Tech. Rep., Dept. of Chemical Materials Engineering, Univ. of Alberta, Edmonton, Canada (2000).

Li, W., and S. Shah, "Structured Residual Vector-Based Approach to Sensor Fault Detection and Isolation," *J. Process. Control*, **12**, 429 (2002).

Li, W., H. Yue, S. Valle-Cervantes, and S. J. Qin, "Recursive PCA for Adaptive Process Monitoring," *J. Process Control*, **10**, 471 (2002).

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

MacGregor, J., "Multivariate Statistical Methods for Monitoring Large Datasets from Chemical Processes," AIChE Meeting, San Francisco, CA (1989).

Negiz, A., and A. Cinar, "Stastical Monitoring of Mulitvarialbe Dynamic Processes with State-Space Models," *AIChE J.*, **43**, 2002 (1997).

Negiz, A., and A. Cinar, "Monitoring of Multivariable Dynamic Processes and Sensor Auditing," *J. Process Control*, **8**, 375 (1998).

Piovoso, M., K. Kosanovich, and R. Pearson, "Monitoring Process Performance in Real-Time," *Proc. Amer. Control Conf.*, Chicago, IL, Chicago, IL, p. 2359 (1992).

Proakis, J., and D. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd ed., Prentice Hall, Upper Saddle River, NJ (1996).

Qin, S. J., and W. Li, "Detection and Identification of Faulty Sensors in Dynamic Processes,"*AIChE J.*, **47**, 1581 (2001).

Wise, B., N. Gallagher, and J. MacGregor, "The Process Chemometrics Approach to Process Monitoring and Fault Detection," *Proc. IFAC Workshop on On-Line Fault Detection and Supervision in the Chemical Process Industries*, International Federation of Automatic Control, Newcastle, U.K. (1995).

Xia, L., and J. Moore, "Recursive Identification of Overparameterized Systems," *IEEE Trans. Automat. Control*, **AC-34**, 327 (1998).

Appendix A: Time-Recursion and Order-Recursion Equations

In the algorithms given below, where unless otherwise stated, $s = 1 \cdots M$, $\eta = 1 \cdots M_n - 1$, and $k >= 2\mu + \text{Int}\,(s + \eta - 1/M)$.

$$\begin{bmatrix} \epsilon_{s,\eta}^{i}(k) \\ \gamma_{s,\eta}^{i}(k) \end{bmatrix} = \begin{cases} \begin{bmatrix} 1 & -\dfrac{\overline{\omega}_{s,\eta}^{i,-\mu}(k)}{\beta_{s+1,\eta-1}^{i}(k)} \\ -\dfrac{\overline{\omega}_{s,\eta}^{i,\mu}(k)}{\alpha_{s,\eta-1}^{i}(k)} & 1 \end{bmatrix} \begin{bmatrix} \epsilon_{s,\eta-1}^{i}(k) \\ \gamma_{s+1,\eta-1}^{i}(k) \end{bmatrix}, & s \neq M \\ \begin{bmatrix} 1 & -\dfrac{\overline{\omega}_{s,\eta}^{i,-\mu}(k)}{\beta_{1,\eta-1}^{i}(k-1)} \\ -\dfrac{\overline{\omega}_{s,\eta}^{i,\mu}(k)}{\overline{\alpha}_{s,\eta-1}^{i}(k)} & 1 \end{bmatrix} \begin{bmatrix} \epsilon_{s,\eta-1}^{i}(k) \\ \gamma_{1,\eta-1}^{i}(k-1) \end{bmatrix}, & s = M \end{cases} \tag{A1}$$

$$\overline{\alpha}^i_{s,\eta}(k) = \begin{cases} \overline{\alpha}^i_{s,\eta-1}(k) - \dfrac{\overline{\omega}^{i,\mu}_{s,\eta}(k)\overline{\omega}^{i,-\mu}_{s,\eta}(k)}{\beta^i_{s+1,\eta-1}(k)}, \; s \neq M \\ \overline{\alpha}^i_{s,\eta-1}(k) - \dfrac{\overline{\omega}^{i,\mu}_{s,\eta}(k)\overline{\omega}^{i,-\mu}_{s,\eta}(k)}{\beta^i_{1,\eta-1}(k-1)}, \; s = M \end{cases} \tag{A2}$$

$$\beta^i_{s,\eta}(k) = \begin{cases} \beta^i_{s+1,\eta-1}(k) - \dfrac{\overline{\omega}^{i,-\mu}_{s,\eta}(k)\overline{\omega}^{i,\mu}_{s,\eta}(k)}{\overline{\alpha}^i_{s,\eta-1}(k)}, \quad s \neq M \\ \beta^i_{1,\eta-1}(k-1) - \dfrac{\overline{\omega}^{i,-\mu}_{s,\eta}(k)\overline{\omega}^{i,\mu}_{s,\eta}(k)}{\alpha^i_{s,\eta-1}(k)}, \quad s = M. \end{cases} \tag{A3}$$

$$\epsilon^{i,\mu}_{s,\eta}(k) = \begin{cases} \epsilon^{i,\mu}_{s,\eta-1}(k) - \dfrac{\overline{\omega}^{i,\mu}_{s,\eta}(k)}{\beta^i_{s+1,\eta-1}(k)} \varsigma^{i,\mu}_{s+1,\eta-1}(k) \quad s \neq M \\ \epsilon^{i,\mu}_{s,\eta-1}(k) - \dfrac{\overline{\omega}^{i,\mu}_{s,\eta+1}(k)}{\beta^i_{1,\eta-1}(k-1)} \varsigma^{i,\mu}_{1,\eta-1}(k-1) \quad s = M \end{cases} \tag{A4}$$

$$\varsigma^{i,\mu}_{s,\eta}(k) = \begin{cases} \varsigma^{i,\mu}_{s+1,\eta-1}(k) - \dfrac{\overline{\omega}^{i,-\mu}_{s,\eta}(k)}{\alpha^i_{s,\eta-1}(k)} \epsilon^{i,\mu}_{s,\eta-1}(k) \quad s \neq M \\ \varsigma^{i,\mu}_{1,\eta-1}(k-1) - \dfrac{\overline{\omega}^{i,-\mu}_{s,\eta}(k)}{\alpha^i_{s,\eta-1}(k)} \epsilon^{i,\mu}_{s,\eta}(k) \quad s = M \end{cases} \tag{A5}$$

$$\epsilon^i_{s,0}(k) = \gamma^i_{s,0}(k) = z^i(kM - s) \tag{A6}$$

$$\epsilon^{i,\mu}_{s,0}(k) = \varsigma^{i,\mu}_{s,0}(k) = z^i[(k-\mu)M - s], \quad s = 1 \cdots M. \tag{A7}$$

$$\overline{\alpha}^i_{s,0}(k) = \beta^i_{s,0}(k) = \sum_{t=-\infty}^{k} \lambda^{k-t} z^i(tM - s) z^i[(t-\mu)M - s] \tag{A8}$$

$$\pi^i_{s,\eta+1}(k) = \pi^i_{s,\eta}(k) + \frac{\gamma^i_{s,\eta}(k)\varsigma^{i,\mu}_{s,\eta}(k)}{\beta^i_{s,\eta}(k)}, \quad \text{with } \pi^i_{s,0}(k) = 0 \tag{A9}$$

$$\pi^i_{s,\eta+1}(k) = \begin{cases} \pi^i_{s+1,\eta}(k) + \dfrac{\epsilon^{i,\mu}_{s,\eta}(k)\epsilon^i_{s,\eta}(k)}{\overline{\alpha}^i_{s,\eta}(k)} \quad s \neq M \\ \pi^i_{1,\eta}(k-1) + \dfrac{\epsilon^{i,\mu}_{s,\eta}(k)\epsilon^i_{s,\eta}(k)}{\overline{\alpha}^i_{s,\eta}(k)} \quad s = M \end{cases} \tag{A10}$$

$$\overline{\alpha}^i_{s,\eta}(k) = \begin{cases} \lambda\overline{\alpha}^i_{s,\eta}(k-1) + \dfrac{\epsilon^i_{s,\eta}(k)\epsilon^{i,\mu}_{s,\eta}(k)}{1 - \pi^i_{s+1,\eta}(k)}, \quad s \neq M, \quad \eta = 0 \cdots M_n - 1 \\ \lambda\overline{\alpha}^i_{s,\eta}(k-1) + \dfrac{\epsilon^i_{s,\eta}(k)\epsilon^{i,\mu}_{s,\eta}(k)}{1 - \pi^i_{1,\eta}(k-1)}, \quad s = M, \quad \eta = 0 \cdots M_n - 1 \end{cases} \tag{A11}$$

$$\beta^i_{s,\eta}(k) = \lambda\beta^i_{s,\eta}(k-1) + \frac{\gamma^i_{s,\eta}(k)\varsigma^{i,\mu}_{s,\eta}(k)}{1 - \pi^i_{s,\eta}(k)}, \quad \eta = 0 \cdots M_n - 1 \tag{A12}$$

$$\overline{\omega}^{i,\mu}_{s,\eta}(k) = \begin{cases} \lambda\overline{\omega}^{i,\mu}_{s,\eta}(k-1) + \dfrac{\epsilon^{i,\mu}_{s,\eta-1}(k)\gamma^i_{s+1,\eta-1}(k)}{1 - \pi^i_{s+1,\eta-1}(k)}, \quad s \neq M \\ \lambda\overline{\omega}^{i,\mu}_{s,\eta}(k-1) + \dfrac{\epsilon^{i,\mu}_{s,\eta-1}(k)\gamma^i_{1,\eta-1}(k)}{1 - \pi^i_{1,\eta-1}(k)}, \quad s = M \end{cases} \tag{A13}$$

$$\overline{\omega}^{i,-\mu}_{s,\eta}(k) = \begin{cases} \lambda\overline{\omega}^{i,-\mu}_{s,\eta}(k-1) + \dfrac{\varsigma^{i,\mu}_{s+1,\eta-1}(k)\epsilon^i_{s,\eta-1}(k)}{1 - \pi^i_{s+1,\eta-1}(k)}, \quad s \neq M \\ \lambda\overline{\omega}^{i,-\mu}_{s,\eta}(k-1) + \dfrac{\varsigma^{i,\mu}_{1,\eta-1}(k-1)\epsilon^i_{s,\eta-1}(k)}{1 - \pi^i_{1,\eta-1}(k-1)}, \quad s = M \end{cases} \tag{A14}$$

Note that the preceding algorithm should be executed for $i = 1$ to $i = m$.

Appendix B: Initialization of the Lattice-Filter Algorithms

We assume that the process data are available only for $k \geq 0$. Therefore, we have to initalize the lattice filtering algorithms at a certain time instant. This can be done by assigning initial values to the quantities $\overline{\alpha}^i_{s,0}(k)$ and $\overline{\omega}^i_{s,\eta}(k)$. To compute $\epsilon^i_{s,\eta}(k)$ and $\lambda^i_{s,\eta}(k)$ for all

$$s = 1 \ldots M, \quad \eta = 1 \ldots M_n - 1, \quad \text{and } k = 0 \ldots \infty$$

the following initialization procedure is suggested.

1. Assign initial values to the quantities

$$\overline{\alpha}^i_{s,0}(-1) \text{ and } \overline{\omega}^{i,\mu}_{s,\eta}\left(\text{Int}\left(\frac{s+\eta-1}{M} - 1\right) + 2\mu\right)$$

$$\text{for } s = 1 \ldots M \text{ and } \eta = 1 \ldots M_n^{-1}$$

2. For $k \geq 0$ and $s = 1 \ldots M$, execute the complete lattice filtering algorithms from $\eta = 1$ until $\text{Int}[(s+\eta-1)/M)] + 2\mu < k$.
3. When $k \geq \text{Int}[(M + M_n - 2)/M)] + 2\mu$, execute the complete algorithms from $\eta = 1$ until $M_n - 1$.

Note that the operator $\text{Int}(\cdot)$ means that the argument has to be converted into an integer.

Manuscript received May 1, 2000, and revision received Sept. 18, 2001.