

Research Article

Integrated Multiscale Latent Variable Regression and Application to Distillation Columns

Muddu Madakyaru,¹ Mohamed N. Nounou,¹ and Hazem N. Nounou²

¹ Chemical Engineering Program, Texas A&M University at Qatar, Doha, Qatar

² Electrical and Computer Engineering Program, Texas A&M University at Qatar, Doha, Qatar

Correspondence should be addressed to Mohamed N. Nounou; mohamed.nounou@qatar.tamu.edu

Received 14 November 2012; Revised 20 February 2013; Accepted 20 February 2013

Academic Editor: Guowei Wei

Copyright © 2013 Muddu Madakyaru et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Proper control of distillation columns requires estimating some key variables that are challenging to measure online (such as compositions), which are usually estimated using inferential models. Commonly used inferential models include latent variable regression (LVR) techniques, such as principal component regression (PCR), partial least squares (PLS), and regularized canonical correlation analysis (RCCA). Unfortunately, measured practical data are usually contaminated with errors, which degrade the prediction abilities of inferential models. Therefore, noisy measurements need to be filtered to enhance the prediction accuracy of these models. Multiscale filtering has been shown to be a powerful feature extraction tool. In this work, the advantages of multiscale filtering are utilized to enhance the prediction accuracy of LVR models by developing an integrated multiscale LVR (IMSLVR) modeling algorithm that integrates modeling and feature extraction. The idea behind the IMSLVR modeling algorithm is to filter the process data at different decomposition levels, model the filtered data from each level, and then select the LVR model that optimizes a model selection criterion. The performance of the developed IMSLVR algorithm is illustrated using three examples, one using synthetic data, one using simulated distillation column data, and one using experimental packed bed distillation column data. All examples clearly demonstrate the effectiveness of the IMSLVR algorithm over the conventional methods.

1. Introduction

In the chemical process industry, models play a key role in various process operations, such as process control, monitoring, and scheduling. For example, the control of a distillation column requires the availability of the distillate and bottom stream compositions. Measuring compositions online is very challenging and costly; therefore, these compositions are usually estimated (using inferential models) from other process variables, which are easier to measure, such as temperature, pressure, flow rates, heat duties, and others. However, there are several challenges that can affect the accuracy of these inferential models, which include the presence of collinearity (or redundancy among the variables) and the presence of measurement noise in the data.

The presence of collinearity, which is due to the large number of variables associated with inferential models, increases the uncertainty about the estimated model parameters and degrades its prediction accuracy. Latent variable

regression (LVR), which is a commonly used framework in inferential modeling, deals with collinearity among the variables by transforming the variables so that most of the data information is captured in a smaller number of variables that can be used to construct the model. In fact, LVR models perform regression on a small number of latent variables that are linear combinations of the original variables. This generally results in well-conditioned models and good predictions [1]. LVR model estimation techniques include principal component regression (PCR) [2, 3], partial least squares (PLS) [2, 4, 5], and regularized canonical correlation analysis (RCCA) [6–9]. PCR is performed in two main steps: transform the input variables using principal component analysis (PCA), and then construct a simple model relating the output to the transformed inputs (principal components) using ordinary least squares (OLS). Thus, PCR completely ignores the output(s) when determining the principal components. Partial least squares (PLS), on the other hand, transform the variables taking the input-output relationship into account

by maximizing the covariance between the output and the transformed input variables. That is why PLS has been widely utilized in practice, such as in the chemical industry to estimate distillation column compositions [10–13]. Other LVR model estimation methods include regularized canonical correlation analysis (RCCA). RCCA is an extension of another estimation technique called canonical correlation analysis (CCA), which determines the transformed input variables by maximizing the correlation between the transformed inputs and the output(s) [6, 14]. Thus, CCA also takes the input-output relationship into account when transforming the variables. CCA, however, requires computing the inverses of the input covariance matrix. Thus, in the case of collinearity among the variables or rank deficiency, regularization of these matrices is performed to enhance the conditioning of the estimated model and, thus, is referred to as regularized CCA (RCCA). Since the covariance and correlation of the transformed variables are related, RCCA reduces to PLS under certain assumptions.

The other challenge in constructing inferential models is the presence of measurement noise in the data. Measured process data are usually contaminated by random and gross errors due to normal fluctuations, disturbances, instrument degradation, and human errors. Such errors mask the important features in the data and degrade the prediction ability of the estimated inferential model. Therefore, measurement noise needs to be filtered for improved model's prediction. Unfortunately, measured data are usually multiscale in nature, which means that they contain features and noise with varying contributions over both time and frequency [15]. For example, an abrupt change in the data spans a wide range in the frequency domain and a small range in the time domain, while a slow change spans a wide range in the time domain and a small range in the frequency domain. Filtering such data using conventional low pass filters, such as the mean filter (MF) or exponentially weighted moving average (EWMA) filter, does not usually provide a good noise-feature separation because these filtering techniques classify noise as high frequency features and filter the data by removing all features having frequencies higher than a defined threshold. Thus, modeling multiscale data requires developing multiscale modeling techniques that can take this multiscale nature of the data into account.

Many investigators have used multiscale techniques to improve the accuracy of estimated empirical models [16–27]. For example, in [17], the authors used multiscale representation of data to design wavelet prefilters for modeling purposes. In [16], on the other hand, the author discussed the advantages of using multiscale representation in empirical modeling, and in [18], he developed a multiscale PCA modeling technique and used it in process monitoring. Also, in [19, 20, 23], the authors used multiscale representation to reduce collinearity and shrink the large variations in FIR model parameters. Furthermore, in [21, 24], multiscale representation was utilized to enhance the prediction and parsimony of fuzzy and ARX models, respectively. In [22], the author extends the classic single-scale system identification tools to the description of multiscale systems. In [25], the authors developed a multiscale latent variable regression (MSLVR)

modeling algorithm by decomposing the input-output data at multiple scales using wavelet and scaling functions and then constructing multiple latent variable regression models at multiple scales using the scaled signal approximations of the data. Note that in this MSLVR approach [25], the LVR models are estimated using only the scaled signals and thus neglect the effect of any significant wavelet coefficients on the model input-output relationship. Later, the same authors extended the same principle to construct nonlinear models using multiscale representation [26]. Finally, in [27], wavelets were used as modulating functions for control-related system identification. Unfortunately, the advantages of multiscale filtering have not been fully utilized to enhance the prediction accuracy of the general class of latent variable regression (LVR) models (e.g., PCR, PLS, and RCCA), which is the focus of this paper.

The objective of this paper is to utilize wavelet-based multiscale filtering to enhance the prediction accuracy of LVR models by developing a modeling technique that integrates multiscale filtering and LVR model estimation. The sought technique should provide improvement over conventional LVR methods as well as those obtained by prefiltering the process data (using low pass or multiscale filters).

The remainder of this paper is organized as follows. In Section 2, a statement of the problem addressed in this work is presented, followed by descriptions of several commonly used LVR model estimation techniques in Section 3. In Section 4, brief descriptions of low pass and multiscale filtering techniques are presented. Then, in Section 5, the advantages of utilizing multiscale filtering in empirical modeling are discussed, followed by a description of an algorithm, called integrated multiscale LVR modeling (IMSLVR), that integrates multiscale filtering and LVR modeling. Then, in Section 6, the performance of the developed IMSLVR modeling technique is assessed through three examples, two simulated examples using synthetic data and distillation column data, and one experimental example using practical packed bed distillation column data. Finally, concluding remarks are presented in Section 7.

2. Problem Statement

This work addresses the problem of enhancing the prediction accuracy of linear inferential models (that can be used to estimate or infer key process variables that are difficult or expensive to measure from more easily measured ones) using multiscale filtering. All variables, inputs and outputs, are assumed to be contaminated with additive zero-mean Gaussian noise. Also, it is assumed that there exists a strong collinearity among the variables. Thus, given noisy measurements of the input and output data, it is desired to construct a linear model with enhanced prediction ability (compared to existing LVR modeling methods) using multiscale data filtering. A general form of a linear inferential model can be expressed as

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \epsilon, \quad (1)$$

where $\mathbf{X} \in \mathbb{R}^{n \times m}$ is the input matrix, $\mathbf{y} \in \mathbb{R}^{n \times 1}$ is the output vector, $\mathbf{b} \in \mathbb{R}^{m \times 1}$ is the unknown model parameter vector, and $\epsilon \in \mathbb{R}^{n \times 1}$ is the model error, respectively.

Multiscale filtering has great feature extraction properties as will be discussed in Sections 4 and 5. However, modeling prefiltered data may result in the elimination of model-relevant information from the filtered input-output data. Therefore, the developed multiscale modeling technique is expected to integrate multiscale filtering and LVR model estimation to enhance the prediction ability of the estimated LVR model. Some of the conventional LVR modeling methods are described next.

3. Latent Variable Regression (LVR) Modeling

One main challenge in developing inferential models is the presence of collinearity among the large number of process variables associated with these models, which affects their prediction ability. Multivariate statistical projection methods such as PCR, PLS, and RCCA can be utilized to deal with this issue by performing regression on a smaller number of transformed variables, called latent variables (or principal components), which are linear combinations of the original variables. This approach, which is called latent variable regression (LVR), generally results in well-conditioned parameter estimates and good model predictions [1].

In this section, descriptions of some of the well-known LVR modeling techniques, which include PCR, PLS, and RCCA, are presented. However, before we describe these techniques, let us introduce some definitions. Let the matrix \mathbf{D} be defined as the augmented scaled input and output data, that is, $\mathbf{D} = [\mathbf{X}\mathbf{y}]$. Note that scaling the data is performed by making each variable (input and output) zero-mean with a unit variance. Then, the covariance of \mathbf{D} can be defined as follows [9]:

$$\begin{aligned} \mathbf{C} &= E(\mathbf{D}\mathbf{D}^T) = E([\mathbf{X}\mathbf{y}]^T [\mathbf{X}\mathbf{y}]) \\ &= \begin{bmatrix} E(\mathbf{X}^T\mathbf{X}) & E(\mathbf{X}^T\mathbf{y}) \\ E(\mathbf{y}^T\mathbf{X}) & E(\mathbf{y}^T\mathbf{y}) \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{\mathbf{X}\mathbf{X}} & \mathbf{C}_{\mathbf{X}\mathbf{y}} \\ \mathbf{C}_{\mathbf{y}\mathbf{X}} & \mathbf{C}_{\mathbf{y}\mathbf{y}} \end{bmatrix}, \end{aligned} \quad (2)$$

where the matrices $\mathbf{C}_{\mathbf{X}\mathbf{X}}$, $\mathbf{C}_{\mathbf{X}\mathbf{y}}$, $\mathbf{C}_{\mathbf{y}\mathbf{X}}$, and $\mathbf{C}_{\mathbf{y}\mathbf{y}}$ are of dimensions $(m \times m)$, $(m \times 1)$, $(1 \times m)$, and (1×1) , respectively.

Since the latent variable model will be developed using transformed (latent) variables, let us define the transformed inputs as follows:

$$\mathbf{z}_i = \mathbf{X}\mathbf{a}_i, \quad (3)$$

where \mathbf{z}_i is the i th latent input variable ($i = 1, \dots, m$), and \mathbf{a}_i is the i th input loading vector, which is of dimension $(m \times 1)$.

3.1. Principal Component Regression (PCR). PCR accounts for collinearity in the input variables by reducing their dimension using principal component analysis (PCA), which utilizes singular value decomposition (SVD) to compute the latent variables or principal components. Then, it constructs a simple linear model between the latent variables and the output

using ordinary least square (OLS) regression [2, 3]. Therefore, PCR can be formulated as two consecutive estimation problems. First, the loading vectors are estimated by maximizing the variance of the estimated principal components as follows:

$$\begin{aligned} \hat{\mathbf{a}}_i &= \arg \max_{\mathbf{a}_i} \text{var}(\mathbf{z}_i) \quad (i = 1, \dots, m), \\ \text{s.t. } &\mathbf{a}_i^T \mathbf{a}_i = 1, \quad \mathbf{z}_i = \mathbf{X}\mathbf{a}_i, \end{aligned} \quad (4)$$

which (because the data are mean centered) can also be expressed in terms of the input covariance matrix $\mathbf{C}_{\mathbf{X}\mathbf{X}}$ as follows:

$$\begin{aligned} \hat{\mathbf{a}}_i &= \arg \max_{\mathbf{a}_i} \mathbf{a}_i^T \mathbf{C}_{\mathbf{X}\mathbf{X}} \mathbf{a}_i \quad (i = 1, \dots, m), \\ \text{s.t. } &\mathbf{a}_i^T \mathbf{a}_i = 1. \end{aligned} \quad (5)$$

The solution of the optimization problem (5) can be obtained using the method of Lagrangian multiplier, which results in the following eigenvalue problem [3, 28]:

$$\mathbf{C}_{\mathbf{X}\mathbf{X}} \hat{\mathbf{a}}_i = \lambda_i \hat{\mathbf{a}}_i, \quad (6)$$

which means that the estimated loading vectors are the eigenvectors of the matrix $\mathbf{C}_{\mathbf{X}\mathbf{X}}$.

Secondly, after the principal components (PCs) are computed, a subset (or all) of these PCs (which correspond to the largest eigenvalues) are used to construct a simple linear model (that relates these PCs to the output) using OLS. Let the subset of PCs used to construct the model be defined as $\mathbf{Z} = [\mathbf{z}_1 \cdots \mathbf{z}_p]$, where $p \leq m$, then the model parameters relating these PCs to the output can be estimated using the following optimization problem:

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} (\|\mathbf{Z}\boldsymbol{\beta} - \mathbf{y}\|_2^2), \quad (7)$$

which has the following closed-form solution,

$$\hat{\boldsymbol{\beta}} = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{y}. \quad (8)$$

Note that if all the estimated principal components are used in constructing the inferential model (i.e., $p = m$), then PCR reduces to OLS. Note also that all principal components in PCR are estimated at the same time (using (6)) and without taking the model output into account. Other methods that take the input-output relationship into consideration when estimating the principal components include partial least squares (PLS) and regularized canonical correlation analysis (RCCA), which are presented next.

3.2. Partial Least Squares (PLS). PLS computes the input loading vectors, \mathbf{a}_i , by maximizing the covariance between the estimated latent variable $\hat{\mathbf{z}}_i$ and model output, \mathbf{y} , that is, [14, 29],

$$\begin{aligned} \hat{\mathbf{a}}_i &= \arg \max_{\mathbf{a}_i} \text{cov}(\mathbf{z}_i, \mathbf{y}), \\ \text{s.t. } &\mathbf{a}_i^T \mathbf{a}_i = 1, \quad \mathbf{z}_i = \mathbf{X}\mathbf{a}_i, \end{aligned} \quad (9)$$

where $i = 1, \dots, p$, $p \leq m$. Since $\mathbf{z}_i = \mathbf{X}\mathbf{a}_i$ and the data are mean centered, (9) can also be expressed in terms of the covariance matrix $\mathbf{C}_{\mathbf{X}\mathbf{y}}$ as follows:

$$\begin{aligned} \hat{\mathbf{a}}_i &= \arg \max_{\mathbf{a}_i} \mathbf{a}_i^T \mathbf{C}_{\mathbf{X}\mathbf{y}}, \\ \text{s.t. } \mathbf{a}_i^T \mathbf{a}_i &= 1. \end{aligned} \quad (10)$$

The solution of the optimization problem (10) can be obtained using the method of Lagrangian multiplier, which leads to the following eigenvalue problem [3, 28]:

$$\mathbf{C}_{\mathbf{X}\mathbf{y}} \mathbf{C}_{\mathbf{y}\mathbf{X}} \hat{\mathbf{a}}_i = \lambda_i^2 \hat{\mathbf{a}}_i \quad (11)$$

which means that the estimated loading vectors are the eigenvectors of the matrix $(\mathbf{C}_{\mathbf{X}\mathbf{y}} \mathbf{C}_{\mathbf{y}\mathbf{X}})$.

Note that PLS utilizes an iterative algorithm [14, 30] to estimate the latent variables used in the model, where one latent variable or principal component is added iteratively to the model. After the inclusion of a latent variable, the input and output residuals are computed and the process is repeated using the residual data until a cross-validation error criterion is minimized [2, 3, 30, 31].

3.3. Regularized Canonical Correlation Analysis (RCCA). RCCA is an extension of a method called canonical correlation analysis (CCA), which was first proposed by Hotelling [6]. CCA reduces the dimension of the model input space by exploiting the correlation among the input and output variables. The assumption behind CCA is that the input and output data contain some joint information that can be represented by the correlation between these variables. Thus, CCA computes the model loading vectors by maximizing the correlation between the estimated principal components and the model output [6–9], that is,

$$\begin{aligned} \hat{\mathbf{a}}_i &= \arg \max_{\mathbf{a}_i} \text{corr}(\mathbf{z}_i, \mathbf{y}), \\ \text{s.t. } \mathbf{z}_i &= \mathbf{X}\mathbf{a}_i, \end{aligned} \quad (12)$$

where $i = 1, \dots, p$, $p \leq m$. Since the correlation between two variables is the covariance divided by the product of the variances of the individual variables, (12) can be written in terms of the covariance between \mathbf{z}_i and \mathbf{y} subject to the following two additional constraints: $\hat{\mathbf{a}}_i^T \mathbf{C}_{\mathbf{X}\mathbf{X}} \hat{\mathbf{a}}_i = 1$ and $\mathbf{C}_{\mathbf{y}\mathbf{y}} = 1$. Thus, the CCA formulation can be expressed as follows:

$$\begin{aligned} \hat{\mathbf{a}}_i &= \arg \max_{\mathbf{a}_i} \text{cov}(\mathbf{z}_i, \mathbf{y}), \\ \text{s.t. } \mathbf{z}_i &= \mathbf{X}\mathbf{a}_i, \quad \mathbf{a}_i^T \mathbf{C}_{\mathbf{X}\mathbf{X}} \mathbf{a}_i = 1. \end{aligned} \quad (13)$$

Note that the constraint $(\mathbf{C}_{\mathbf{y}\mathbf{y}} = 1)$ is omitted from (13) because it is satisfied by scaling the data to have zero-mean and unit variance as described in Section 3. Since the data are mean centered, (13) can be written in terms of the covariance matrix $\mathbf{C}_{\mathbf{X}\mathbf{y}}$ as follows:

$$\begin{aligned} \hat{\mathbf{a}}_i &= \arg \max_{\mathbf{a}_i} \mathbf{a}_i^T \mathbf{C}_{\mathbf{X}\mathbf{y}}, \\ \text{s.t. } \mathbf{a}_i^T \mathbf{C}_{\mathbf{X}\mathbf{X}} \mathbf{a}_i &= 1. \end{aligned} \quad (14)$$

The solution of the optimization problem (14) can be obtained using the method of Lagrangian multiplier, which leads to the following eigenvalue problem [14, 28]:

$$\mathbf{C}_{\mathbf{X}\mathbf{X}}^{-1} \mathbf{C}_{\mathbf{X}\mathbf{y}} \mathbf{C}_{\mathbf{y}\mathbf{X}} \hat{\mathbf{a}}_i = \lambda_i^2 \hat{\mathbf{a}}_i, \quad (15)$$

which means that the estimated loading vector is the eigenvector of the matrix $\mathbf{C}_{\mathbf{X}\mathbf{X}}^{-1} \mathbf{C}_{\mathbf{X}\mathbf{y}} \mathbf{C}_{\mathbf{y}\mathbf{X}}$.

Equation (15) shows that CCA requires inverting the matrix $\mathbf{C}_{\mathbf{X}\mathbf{X}}$ to obtain the loading vector, \mathbf{a}_i . In the case of collinearity in the model input space, the matrix $\mathbf{C}_{\mathbf{X}\mathbf{X}}$ becomes nearly singular, which results in poor estimation of the loading vectors, and thus a poor model. Therefore, a regularized version of CCA (called RCCA) has been developed to account for this drawback of CCA [14]. The formulation of RCCA can be expressed as follows:

$$\begin{aligned} \hat{\mathbf{a}}_i &= \arg \max_{\mathbf{a}_i} \mathbf{a}_i^T \mathbf{C}_{\mathbf{X}\mathbf{y}}, \\ \text{s.t. } \mathbf{a}_i^T ((1 - \tau_a) \mathbf{C}_{\mathbf{X}\mathbf{X}} + \tau_a \mathbf{I}) \mathbf{a}_i &= 1. \end{aligned} \quad (16)$$

The solution of the optimization problem (16) can be obtained using the method of Lagrangian multiplier, which leads to the following eigenvalue problem [14]:

$$[(1 - \tau_a) \mathbf{C}_{\mathbf{X}\mathbf{X}} + \tau_a \mathbf{I}]^{-1} \mathbf{C}_{\mathbf{X}\mathbf{y}} \mathbf{C}_{\mathbf{y}\mathbf{X}} \hat{\mathbf{a}}_i = \lambda_i^2 \hat{\mathbf{a}}_i, \quad (17)$$

which means that the estimated loading vectors are the eigenvectors of the matrix $([(1 - \tau_a) \mathbf{C}_{\mathbf{X}\mathbf{X}} + \tau_a \mathbf{I}]^{-1} \mathbf{C}_{\mathbf{X}\mathbf{y}} \mathbf{C}_{\mathbf{y}\mathbf{X}})$. Note from (17) that RCCA deals with possible collinearity in the model input space by inverting a weighted sum of the matrix $\mathbf{C}_{\mathbf{X}\mathbf{X}}$ and the identity matrix, that is, $[(1 - \tau_a) \mathbf{C}_{\mathbf{X}\mathbf{X}} + \tau_a \mathbf{I}]$, instead of inverting the matrix $\mathbf{C}_{\mathbf{X}\mathbf{X}}$ itself. However, this requires knowledge of the weighting or regularization parameter τ_a . We know, however, that when $\tau_a = 0$, the RCCA solution (17) reduces to the CCA solution (15), and when $\tau_a = 1$, the RCCA solution (17) reduces to the PLS solution (11) since $\mathbf{C}_{\mathbf{y}\mathbf{y}}$ is a scalar.

3.3.1. Optimizing the RCCA Regularization Parameter. The above discussion shows that depending on the value of τ_a , where $0 \leq \tau_a \leq 1$, RCCA provides a solution that converges to CCA or PLS at the two end points, 0 or 1, respectively. In [14], it has been shown that RCCA can provide better results than PLS for some intermediate values of τ_a between 0 and 1. Therefore, in this section, we propose to optimize the performance of RCCA by optimizing its regularization parameter by solving the following nested optimization problem to find the optimum value of τ_a :

$$\begin{aligned} \hat{\tau}_a &= \arg \min_{\tau_a} (\mathbf{y} - \hat{\mathbf{y}})^T (\mathbf{y} - \hat{\mathbf{y}}), \\ \text{s.t. } \hat{\mathbf{y}} &= \text{RCCA model prediction.} \end{aligned} \quad (18)$$

The inner loop of the optimization problem shown in (18) solves for the RCCA model prediction given the value of the regularization parameter τ_a , and the outer loop selects the value of τ_a that provides the least cross-validation mean square error using unseen testing data.

Note that RCCA solves for the latent variable regression model in an iterative fashion similar to PLS, where one latent variable is estimated in each iteration [14]. Then, the contributions of the latent variable and its corresponding model prediction are subtracted from the input and output data, and the process is repeated using the residual data until an optimum number of principal components or latent variables are used according to some cross-validation error criterion.

4. Data Filtering

In this section, brief descriptions of some of the filtering techniques which will be used later to enhance the prediction of LVR models are presented. These techniques include linear (or low pass) as well as multiscale filtering techniques.

4.1. Linear Data Filtering. Linear filtering techniques filter the data by computing a weighted sum of previous measurements in a window of finite or infinite length and are called finite impulse response (FIR) and infinite impulse response (IIR) filters. A linear filter can be written as follows:

$$\hat{y}_k = \sum_{i=0}^{N-1} w_i y_{k-i}, \quad (19)$$

where $\sum_i w_i = 1$, and N is the filter length. Well-known FIR and IIR filters include the mean filter (MF) and the exponentially weighted moving average (EWMA) filter, respectively. The mean filter uses equal weights, that is, $w_i = 1/N$, while the exponentially weighted moving average (EWMA) filter averages all the previous measurements. The EWMA filter can also be implemented recursively as follows:

$$\hat{y}_k = \alpha y_k + (1 - \alpha) \hat{y}_{k-1}, \quad (20)$$

where y_k and \hat{y}_k are the measured and filtered data samples at time step (k). The parameter α is an adjustable smoothing parameter lying between 0 and 1, where a value of 1 corresponds to no filtering and a value of zero corresponds to keeping only the first measured point. A more detailed discussion of different types of filters is presented in [32].

In linear filtering, the basis functions representing raw measured data have a temporal localization equal to the sampling interval. This means that linear filters are single scale in nature since all the basis functions have the same fixed time-frequency localization. Consequently, these methods face a tradeoff between accurate representation of temporally localized changes and efficient removal of temporally global noise [33]. Therefore, simultaneous noise removal and accurate feature representation of measured signals containing multiscale features cannot be effectively achieved by single-scale filtering methods [33]. Enhanced denoising can be achieved using multiscale filtering as will be described next.

4.2. Multiscale Data Filtering. In this section, a brief description of multiscale filtering is presented. However, since multiscale filtering relies on multiscale representation of data using wavelets and scaling functions, a brief introduction to multiscale representation is presented first.

4.2.1. Multiscale Representation of Data. Any square-integrable signal (or data vector) can be represented at multiple scales by expressing the signal as a superposition of wavelets and scaling functions, as shown in Figure 1. The signals in Figures 1(b), 1(d), and 1(f) are at increasingly coarser scales compared to the original signal shown in Figure 1(a). These scaled signals are determined by filtering the data using a low pass filter of length r , $\mathbf{h}_f = [h_1, h_2, \dots, h_r]$, which is equivalent to projecting the original signal on a set of orthonormal scaling functions of the form

$$\phi_{jk}(t) = \sqrt{2^{-j}} \phi(2^{-j}t - k). \quad (21)$$

On the other hand, the signals in Figures 1(c), 1(e), and 1(g), which are called the detail signals, capture the details between any scaled signal and the scaled signal at the finer scale. These detailed signals are determined by projecting the signal on a set of wavelet basis functions of the form

$$\psi_{jk}(t) = \sqrt{2^{-j}} \psi(2^{-j}t - k) \quad (22)$$

or equivalently by filtering the scaled signal at the finer scale using a high pass filter of length r , $\mathbf{g}_f = [g_1, g_2, \dots, g_r]$, that is derived from the wavelet basis functions. Therefore, the original signal can be represented as the sum of all detailed signals at all scales and the scaled signal at the coarsest scale as follows:

$$x(t) = \sum_{k=1}^{n2^{-j}} \mathbf{a}_{jk} \phi_{jk}(t) + \sum_{j=1}^J \sum_{k=1}^{n2^{-j}} \mathbf{d}_{jk} \psi_{jk}(t), \quad (23)$$

where j , k , J , and n are the dilation parameter, translation parameter, maximum number of scales (or decomposition depth), and the length of the original signal, respectively [27, 34–36].

Fast wavelet transform algorithms with $O(n)$ complexity for a discrete signal of dyadic length have been developed [37]. For example, the wavelet and scaling function coefficients at a particular scale (j), \mathbf{a}_j and \mathbf{d}_j , can be computed in a compact fashion by multiplying the scaling coefficient vector at the finer scale, \mathbf{a}_{j-1} , by the matrices \mathbf{H}_j and \mathbf{G}_j , respectively, that is,

$$\mathbf{a}_j = \mathbf{H}_j \mathbf{a}_{j-1}, \quad \mathbf{d}_j = \mathbf{G}_j \mathbf{a}_{j-1}, \quad (24)$$

where,

$$\mathbf{H}_j = \begin{bmatrix} h_1 & \cdot & h_r & \cdot & \cdot \\ 0 & h_1 & \cdot & h_r & 0 \\ 0 & 0 & \cdot & \cdot & \cdot \\ 0 & 0 & h_1 & \cdot & h_r \end{bmatrix}_{n2^j \times n2^j}, \quad \mathbf{G}_j = \begin{bmatrix} g_1 & \cdot & g_r & \cdot & \cdot \\ 0 & g_1 & \cdot & g_r & 0 \\ 0 & 0 & \cdot & \cdot & \cdot \\ 0 & 0 & g_1 & \cdot & g_r \end{bmatrix}_{n2^j \times n2^j}. \quad (25)$$

Note that the length of the scaled and detailed signals decreases dyadically at coarser resolutions (higher j). In other words, the length of scaled signal at scale (j) is half the length of scaled signal at the finer scale ($j - 1$). This is due to downsampling, which is used in discrete wavelet transform.

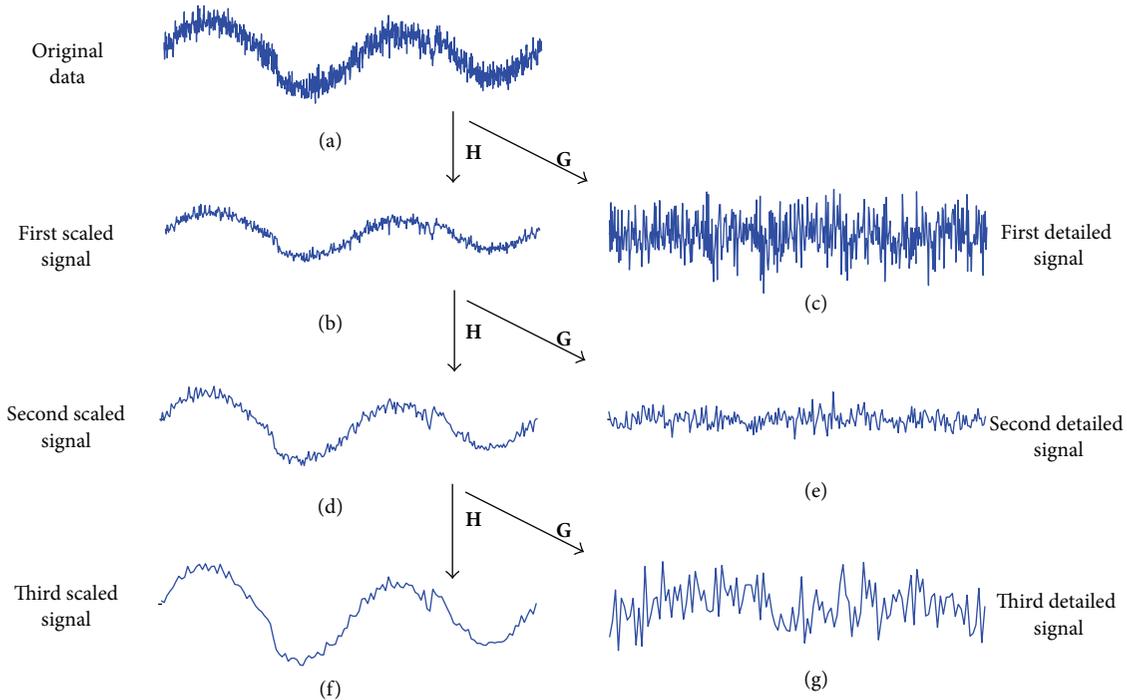


FIGURE 1: Multiscale decomposition of a heavy-sine signal using Haar.

4.2.2. Multiscale Data Filtering Algorithm. Multiscale filtering using wavelets is based on the observation that random errors in a signal are present over all wavelet coefficients while deterministic changes get captured in a small number of relatively large coefficients [16, 38–41]. Thus, stationary Gaussian noise may be removed by a three-step method [40].

- (i) Transform the noisy signal into the time-frequency domain by decomposing the signal on a selected set of orthonormal wavelet basis functions.
- (ii) Threshold the wavelet coefficients by suppressing any coefficients smaller than a selected threshold value.
- (iii) Transform the thresholded coefficients back into the original time domain.

Donoho and coworkers have studied the statistical properties of wavelet thresholding and have shown that for a noisy signal of length n , the filtered signal will have an error within $O(\log n)$ of the error between the noise-free signal and the signal filtered with a priori knowledge of the smoothness of the underlying signal [39].

Selecting the proper value of the threshold is a critical step in this filtering process, and several methods have been devised. For good visual quality of the filtered signal, the VisuShrink method determines the threshold as [42]

$$t_j = \sigma_j \sqrt{2 \log n}, \quad (26)$$

where n is the signal length and σ_j is the standard deviation of the errors at scale j , which can be estimated from the wavelet coefficients at that scale using the following relation:

$$\sigma_j = \frac{1}{0.6745} \text{median} \{ |d_{jk}| \}. \quad (27)$$

Other methods for determining the value of the threshold are described in [43].

5. Multiscale LVR Modeling

In this section, multiscale filtering will be utilized to enhance the prediction accuracy of various LVR modeling techniques in the presence of measurement noise in the data. It is important to note that in practical process data, features and noise span wide ranges over time and frequency. In other words, features in the input-output data may change at a high frequency over a certain time span, but at a much lower frequency over a different time span. Also, noise (especially colored or correlated) may have varying frequency contents over time. In modeling such multiscale data, the model estimation technique should be capable of extracting the important features in the data and removing the undesirable noise and disturbance to minimize the effect of these disturbances on the estimated model.

5.1. Advantages of Multiscale Filtering in LVR Modeling. Since practical process data are usually multiscale in nature, modeling such data requires a multiscale modeling technique that accounts for this type of data. Below is a description of some of the advantages of multiscale filtering in LVR model estimation [44].

- (i) The presence of noise in measured data can considerably affect the accuracy of estimated LVR models. This effect can be greatly reduced by filtering the data using wavelet-based multiscale filtering, which provides effective separation of noise from important features to improve the quality of the estimated models. This noise-feature separation can be visually seen from Figure 1, which shows that the scaled signals are less noise corrupted at coarser scales.
- (ii) Another advantage of multiscale representation is that correlated noise (within each variable) gets approximately decorrelated at multiple scales. Correlated (or colored) noise arises in situations where the source of error is not completely independent and random, such as malfunctioning sensors or erroneous sensor calibration. Having correlated noise in the data makes modeling more challenging because such noise is interpreted as important features in the data, while it is in fact noise. This property of multiscale representation is really useful in practice, where measurement errors are not always random [33].

These advantages will be utilized to enhance the accuracy of LVR models by developing an algorithm that integrates multiscale filtering and LVR model estimation as described next.

5.2. Integrated Multiscale LVR (IMSLVR) Modeling. The idea behind the developed integrated multiscale LVR (IMSLVR) modeling algorithm is to combine the advantages of multiscale filtering and LVR model estimation to provide inferential models with improved predictions. Let the time domain input and output data be \mathbf{X} and \mathbf{y} , and let the filtered data (using the multiscale filtering algorithm described in Section 4.2.2) at a particular scale (j) be \mathbf{X}_j and \mathbf{y}_j ; then the inferential model (which is estimated using these filtered data) can be expressed as follows:

$$\mathbf{y}_j = \mathbf{X}_j \mathbf{b}_j + \epsilon_j, \quad (28)$$

where $\mathbf{X}_j \in \mathbb{R}^{n \times m}$ is the filtered input data matrix at scale (j), $\mathbf{y}_j \in \mathbb{R}^{n \times 1}$ is the filtered output vector at scale (j), $\mathbf{b}_j \in \mathbb{R}^{m \times 1}$ is the estimated model parameter vector using the filtered data at scale (j), and $\epsilon_j \in \mathbb{R}^{n \times 1}$ is the model error when the filtered data at scale (j) are used, respectively.

Before we present the formulations of the LVR modeling techniques using the multiscale filtered data, let us define the following. Let the matrix \mathbf{D}_j be defined as the augmented scaled and filtered input and output data, that is, $\mathbf{D}_j = [\mathbf{X}_j \mathbf{y}_j]$. Then, the covariance of \mathbf{D}_j can be defined as follows [9]:

$$\mathbf{C}_j = E(\mathbf{D}_j \mathbf{D}_j^T) = E\left([\mathbf{X}_j \mathbf{y}_j]^T [\mathbf{X}_j \mathbf{y}_j]\right) = \begin{bmatrix} \mathbf{C}_{\mathbf{X}_j \mathbf{X}_j} & \mathbf{C}_{\mathbf{X}_j \mathbf{y}_j} \\ \mathbf{C}_{\mathbf{y}_j \mathbf{X}_j} & \mathbf{C}_{\mathbf{y}_j \mathbf{y}_j} \end{bmatrix}. \quad (29)$$

Also, since the LVR models are developed using transformed variables, the transformed input variables using the filtered inputs at scale (j) can be expressed as follows:

$$\mathbf{z}_{i,j} = \mathbf{X}_j \mathbf{a}_{i,j}, \quad (30)$$

where $\mathbf{z}_{i,j}$ is the i th latent input variable ($i = 1, \dots, m$) and $\mathbf{a}_{i,j}$ is the i th input loading vector which is estimated using the filtered data at scale (j) using any of the LVR modeling techniques, that is, PCR, PLS, or RCCA. Thus, the LVR model estimation problem (using the multiscale filtered data at scale (j)) can be formulated as follows.

5.2.1. LVR Modeling Using Multiscale Filtered Data. The PCR model can be estimated using the multiscale filtered data at scale (j) as follows:

$$\begin{aligned} \hat{\mathbf{a}}_{i,j} &= \arg \max_{\mathbf{a}_{i,j}} \mathbf{a}_{i,j}^T \mathbf{C}_{\mathbf{X}_j \mathbf{X}_j} \mathbf{a}_{i,j} \quad (i = 1, \dots, m, j = 0, \dots, J), \\ \text{s.t. } &\mathbf{a}_{i,j}^T \mathbf{a}_{i,j} = 1. \end{aligned} \quad (31)$$

Similarly, the PLS model can be estimated using the multiscale filtered data at scale (j) as follows:

$$\begin{aligned} \hat{\mathbf{a}}_{i,j} &= \arg \max_{\mathbf{a}_{i,j}} \mathbf{a}_{i,j}^T \mathbf{C}_{\mathbf{X}_j \mathbf{y}_j} \quad (i = 1, \dots, m, j = 0, \dots, J), \\ \text{s.t. } &\mathbf{a}_{i,j}^T \mathbf{a}_{i,j} = 1. \end{aligned} \quad (32)$$

And finally, the RCCA model can be estimated using the multiscale filtered data at scale (j) as follows:

$$\begin{aligned} \hat{\mathbf{a}}_{i,j} &= \arg \max_{\mathbf{a}_{i,j}} \mathbf{a}_{i,j}^T \mathbf{C}_{\mathbf{X}_j \mathbf{y}_j} \quad (i = 1, \dots, m, j = 0, \dots, J), \\ \text{s.t. } &\mathbf{a}_{i,j}^T \left((1 - \tau_{a,j}) \mathbf{C}_{\mathbf{X}_j \mathbf{X}_j} + \tau_{a,j} \mathbf{I} \right) \mathbf{a}_{i,j} = 1. \end{aligned} \quad (33)$$

5.2.2. Integrated Multiscale LVR Modeling Algorithm. It is important to note that multiscale filtering enhances the quality of the data and the accuracy of the LVR models estimated using these data. However, filtering the input and output data a priori without taking the relationship between these two data sets into account may result in the removal of features that are important to the model. Thus, multiscale filtering needs to be integrated with LVR model for proper noise removal. This is what is referred to as integrated multiscale LVR (IMSLVR) modeling. One way to accomplish this integration between multiscale filtering and LVR modeling is using the following IMSLVR modeling algorithm which is schematically illustrated in Figure 2:

- (i) split the data into two sets: training and testing,
- (ii) scale the training and testing data sets,
- (iii) filter the input and output training data at different scales (decomposition depths) using the algorithm described in Section 4.2.2,
- (iv) using the filtered training data from each scale, construct an LVR model. The number of principal components is optimized using cross-validation,
- (v) use the estimated model from each scale to predict the output for the testing data, and compute the cross-validation mean square error,

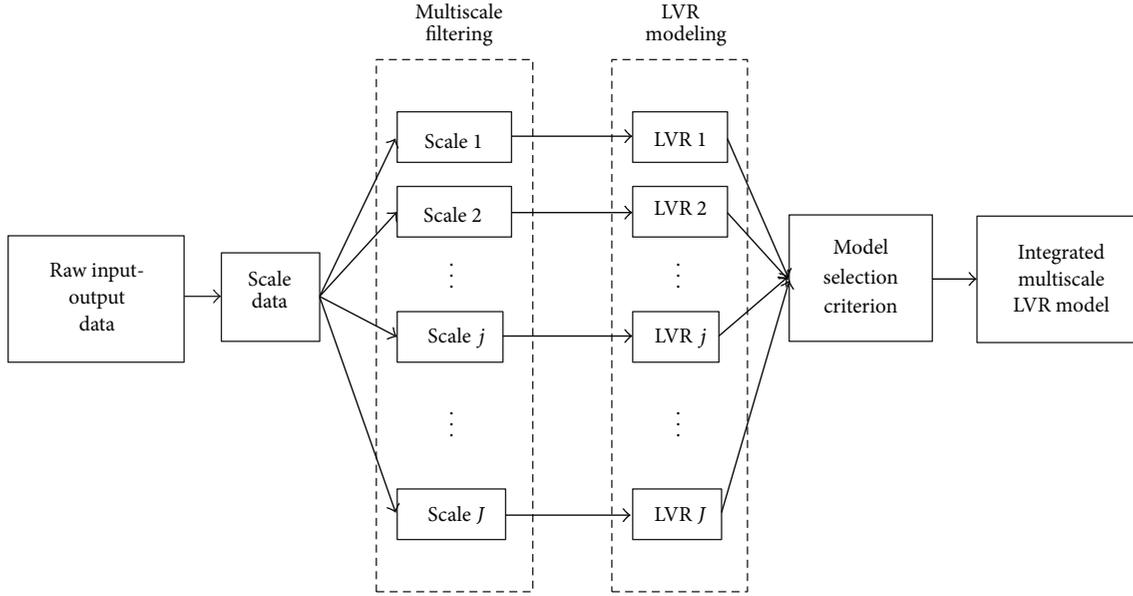


FIGURE 2: A schematic diagram of the integrated multiscale LVR (IMSLVR) modeling algorithm.

- (vi) select the LVR with the least cross-validation mean square error as the IMSLVR model.

6. Illustrative Examples

In this section, the performances of the IMSLVR modeling algorithm described in Section 5.2.2 is illustrated and compared with those of the conventional LVR modeling methods as well as the models obtained by prefiltering the data (using either multiscale filtering or low pass filtering). This comparison is performed through three examples. The first two examples are simulated examples, one using synthetic data and the other using simulated distillation column data. The third example is a practical example that uses experimental packed bed distillation column data. In all examples, the estimated models are optimized and compared using cross-validation, by minimizing the output prediction mean square error (MSE) using unseen testing data as follow:

$$\text{MSE} = \frac{1}{N} \sum_{k=1}^n (y(k) - \hat{y}(k))^2, \quad (34)$$

where $y(k)$ and $\hat{y}(k)$ are the measured and predicted outputs at time step (k), and n is the total number of testing measurements. Also, the number of retained latent variables (or principal components) by the various LVR modeling techniques (RCCA, PLS, and PCR) is optimized using cross-validation. Note that the data (inputs and output) are scaled (by subtracting the mean and dividing by the standard deviation) before constructing the LVR models to enhance their prediction abilities.

6.1. Example 1: Inferential Modeling of Synthetic Data. In this example, the performances of the various LVR modeling

techniques are compared by modeling synthetic data consisting of ten input variables and one output variable.

6.1.1. Data Generation. The data are generated as follows. The first two input variables are “block” and “heavy-sine” signals, and the other input variables are computed as linear combinations of the first two inputs as follows:

$$\begin{aligned} \mathbf{x}_3 &= \mathbf{x}_1 + \mathbf{x}_2, \\ \mathbf{x}_4 &= 0.3\mathbf{x}_1 + 0.7\mathbf{x}_2, \\ \mathbf{x}_5 &= 0.3\mathbf{x}_3 + 0.2\mathbf{x}_4, \\ \mathbf{x}_6 &= 2.2\mathbf{x}_1 - 1.7\mathbf{x}_3, \\ \mathbf{x}_7 &= 2.1\mathbf{x}_6 + 1.2\mathbf{x}_5, \\ \mathbf{x}_8 &= 1.4\mathbf{x}_2 - 1.2\mathbf{x}_7, \\ \mathbf{x}_9 &= 1.3\mathbf{x}_2 + 2.1\mathbf{x}_1, \\ \mathbf{x}_{10} &= 1.3\mathbf{x}_6 - 2.3\mathbf{x}_9, \end{aligned} \quad (35)$$

which means that the input matrix \mathbf{X} is of rank 2. Then, the output is computed as a weighed sum of all inputs as follows:

$$\mathbf{y} = \sum_{i=1}^{10} b_i \mathbf{x}_i, \quad (36)$$

where $b_i = \{0.07, 0.03, -0.05, 0.04, 0.02, -1.1, -0.04, -0.02, 0.01, -0.03\}$, for $i = 1, \dots, 10$. The total number of generated data samples is 512. All variables, inputs and output, which are assumed to be noise-free, are then contaminated with additive zero-mean Gaussian noise. Different levels of noise, which correspond to signal-to-noise ratios (SNR) of 5, 10, and 20, are used to illustrate the performances of the various

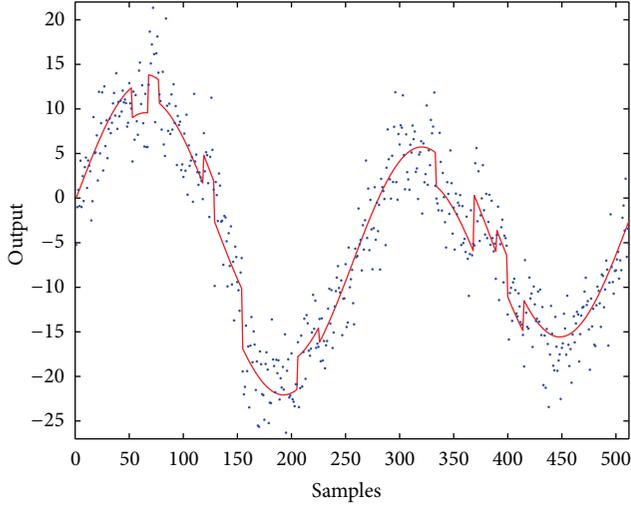


FIGURE 3: Sample output data set used in example 1 for the case where $\text{SNR} = 10$ (solid line: noise-free data; dots: noisy data).

methods at different noise contributions. The SNR is defined as the variance of the noise-free data divided by the variance of the contaminating noise. A sample of the output data, where $\text{SNR} = 10$, is shown in Figure 3.

6.1.2. Selection of Decomposition Depth and Optimal Filter Parameters. The decomposition depth used in multiscale filtering and the parameters of the low pass filters (i.e., the length of the mean filter and the value of the smoothing parameter α) are optimized using a cross-validation criterion, which was proposed in [43]. The idea here is to split the data into two sets: odd (y_o) and even (y_e); filter the odd set, compute estimates of the even numbered data from the filtered odd data by averaging the two adjacent filtered samples, that is, $\hat{y}_{e,i} = (1/2)(\hat{y}_{o,i} + \hat{y}_{o,i+1})$, and then compute the cross-validation MSE (CVMSE) with respect to the even data samples as follows:

$$\text{CVMSE}_{y_e} = \sum_{i=1}^{N/2} (y_{e,i} - \hat{y}_{e,i})^2. \quad (37)$$

The same process is repeated using the even numbered samples as the training data, and then the optimum filter parameters are selected by minimizing the sum of cross-validation mean squared errors using both the odd and even data samples.

6.1.3. Simulation Results. In this section, the performance of the IMSLVR modeling algorithm is compared to those of the conventional LVR algorithms (RCCA, PLS, and PCR) and those obtained by prefiltering the data using multiscale filtering, mean filtering (MF), and EWMA filtering. In multiscale filtering, the Daubechies wavelet filter of order three is used, and the filtering parameters for all filtering techniques are optimized using cross-validation. To obtain statistically valid conclusions, a Monte Carlo simulation using 1000 realizations is performed, and the results are shown in Table 1.

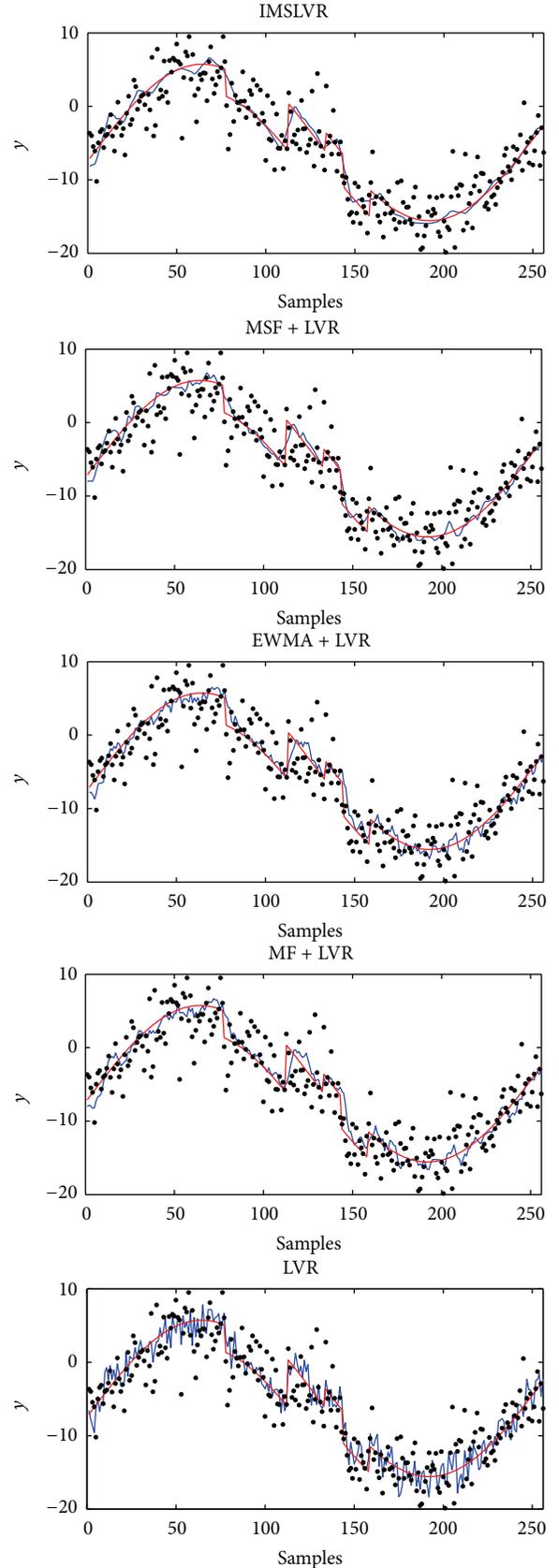


FIGURE 4: Comparison of the model predictions using the various LVR (RCCA) modeling techniques in example 1 for the case where $\text{SNR} = 10$ (solid blue line: model prediction; solid red line: noise-free data; black dots: noisy data).

The results in Table 1 clearly show that modeling prefiltered data (using multiscale filtering (MSF+LVR), EWMA filtering (EWMA+LVR), or mean filtering (MF+LVR)) provides a significant improvement over the conventional LVR modeling techniques. This advantage is much clearer for multiscale filtering over the single-scale (low pass) filtering techniques. However, the IMSLVR algorithm provides a further improvement over multiscale prefiltering (MSF+LVR) for all noise levels. This is because the IMSLVR algorithm integrates modeling and feature extraction to retain features in the data that are important to the model, which improves the model prediction ability. Finally, the results in Table 1 also show that the advantages of the IMSLVR algorithm are clearer for larger noise contents, that is, smaller SNR. As an example, the performances of all estimated models using RCCA are demonstrated in Figure 4 for the case where $SNR = 10$, which clearly shows the advantages of IMSLVR over other LVR modeling techniques.

6.1.4. Effect of Wavelet Filter on Model Prediction. The choice of the wavelet filter has a great impact on the performance of the estimated model using the IMSLVR modeling algorithm. To study the effect of the wavelet filter on the performance of the estimated models, in this example, we repeated the simulations using different wavelet filters (Haar, Daubechies second and third order filters) and results of a Monte Carlo simulation using 1000 realizations are shown in Figure 5. The simulation results clearly show that the Daubechies third order filter is the best filter for this example, which makes sense because it is smoother than the other two filters, and thus it fits the nature of the data better.

6.2. Example 2: Inferential Modeling of Distillation Column Data. In this example, the prediction abilities of the various modeling techniques (i.e., IMSLVR, MSF+LVR, EWMA+LVR, MF+LVR, and LVR) are compared through their application to model the distillate and bottom stream compositions of a distillation column. The dynamic operation of the distillation column, which consists of 32 theoretical stages (including the reboiler and a total condenser), is simulated using Aspen Tech 7.2. The feed stream, which is a binary mixture of propane and isobutene, enters the column at stage 16 as a saturated liquid having a flow rate of 1 kmol/s, a temperature of 322 K, and compositions of 40 mole% propane and 60 mole% isobutene. The nominal steady state operating conditions of the column are presented in Table 2.

6.2.1. Data Generation. The data used in this modeling problem are generated by perturbing the flow rates of the feed and the reflux streams from their nominal operating values. First, step changes of magnitudes $\pm 2\%$ in the feed flow rate around its nominal condition are introduced, and in each case, the process is allowed to settle to a new steady state. After attaining the nominal conditions again, similar step changes of magnitudes $\pm 2\%$ in the reflux flow rate around its nominal condition are introduced. These perturbations are used to generate training and testing data (each consisting of 512 data points) to be used in developing the various models. These

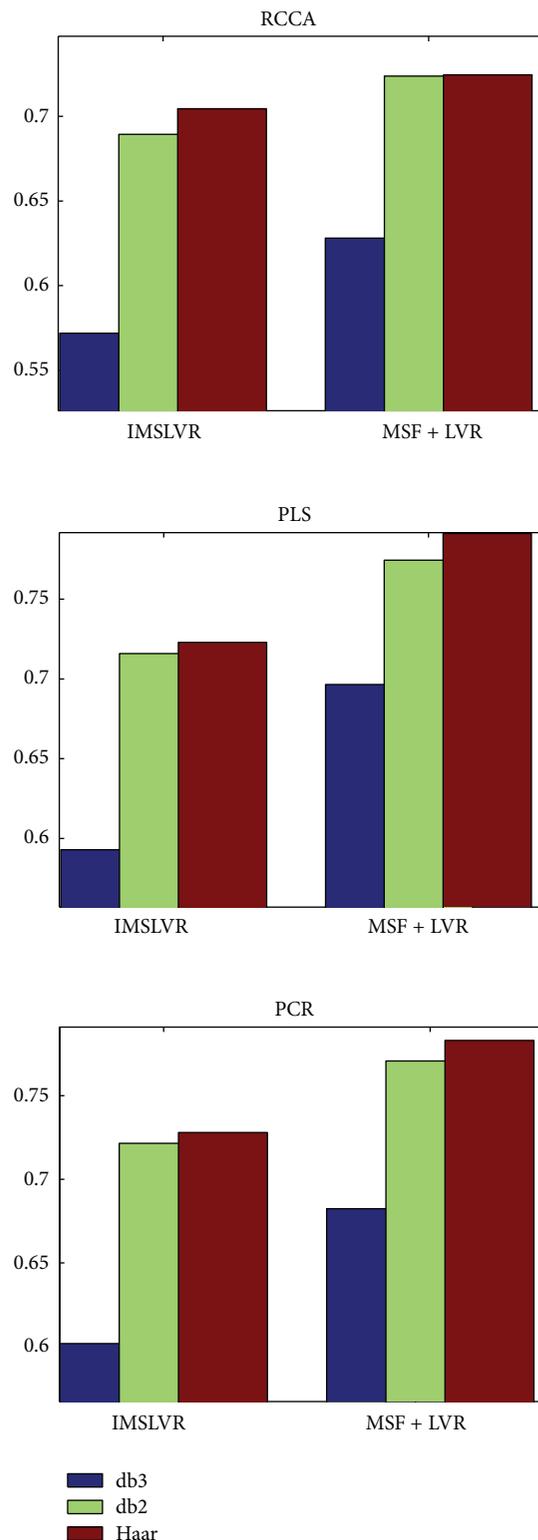


FIGURE 5: Comparison of the MSEs for various wavelet filters in example 1 for the case where $SNR = 10$.

perturbations (in the training and testing data sets) are shown in Figures 6(e), 6(f), 6(g), and 6(h).

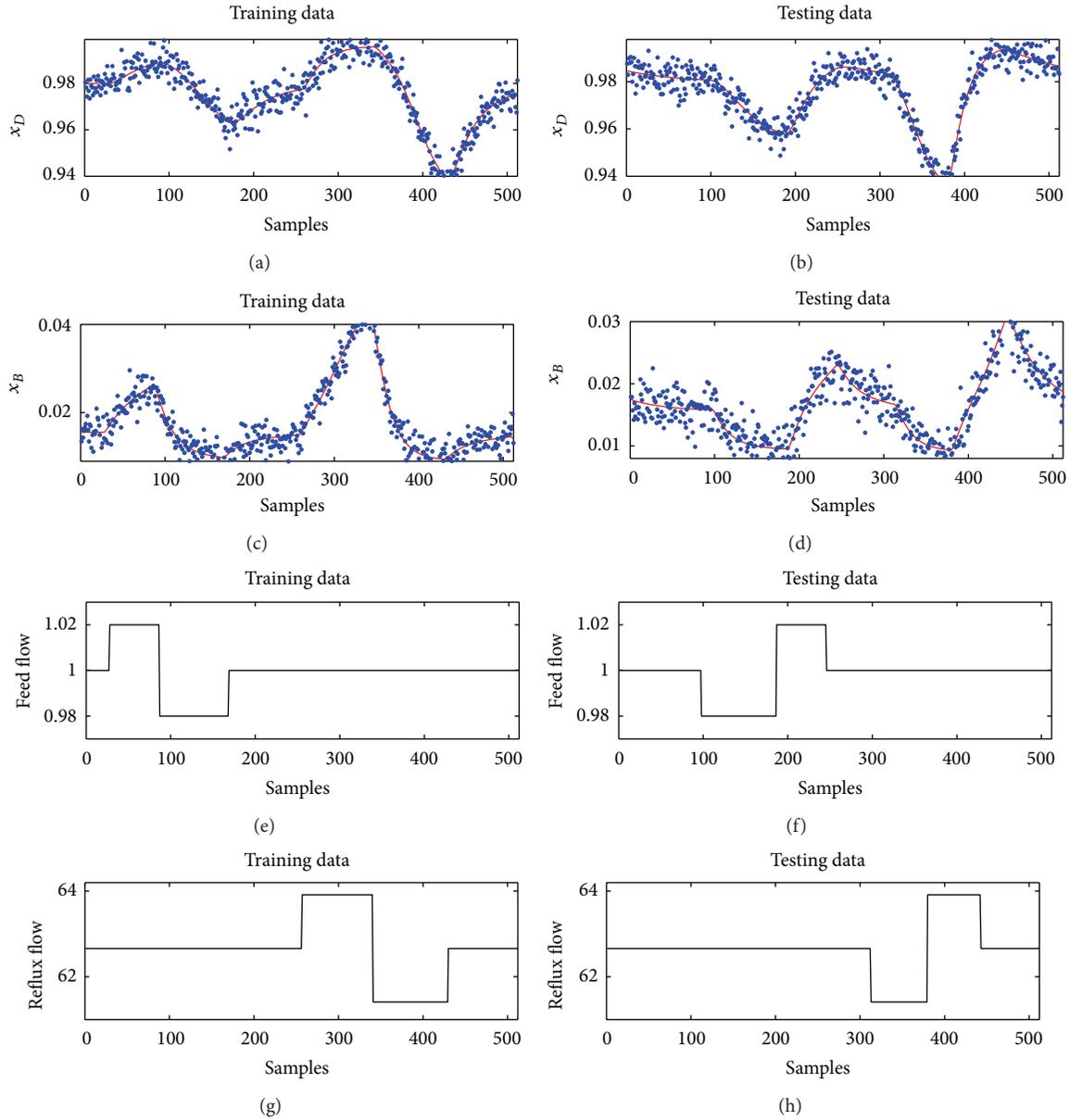


FIGURE 6: The dynamic input-output data used for training and testing the models in the simulated distillation column example for the case where the noise SNR = 10 (solid red line: noise-free data; blue dots: noisy data).

In this simulated modeling problem, the input variables consist of ten temperatures at different trays of the column, in addition to the flow rates of the feed and reflux streams. The output variables, on the other hand, are the compositions of the light component (propane) in the distillate and the bottom streams (i.e., x_D and x_B , resp.). The dynamic temperature and composition data generated using the Aspen simulator (due to the perturbations in the feed and reflux flow rates) are assumed to be noise-free, which are then contaminated with zero-mean Gaussian noise. To assess the robustness of the various modeling techniques to different noise contributions, different levels of noise (which correspond to signal-to-noise ratios of 5, 10, and 20) are used. Sample training and testing

data sets showing the effect of the perturbations on the column compositions are shown in Figures 6(a), 6(b), 6(c), and 6(d) for the case where the signal-to-noise ratio is 10.

6.2.2. Simulation Results. In this section, the performance of the IMSLVR algorithm is compared to the conventional LVR models as well as the models estimated using prefiltered data. To obtain statistically valid conclusions, a Monte Carlo simulation of 1000 realizations is performed, and the results are presented in Tables 3 and 4 for the estimation of top and bottom distillation column compositions, that is, x_D and x_B , respectively. As in the first example, the results in both

TABLE 1: Comparison of the Monte Carlo MSEs for the various modeling techniques in example 1.

| Model type | IMSLVR | MSF+LVR | EWMA+LVR | MF+LVR | LVR |
|------------|--------|---------|----------|--------|--------|
| SNR = 5 | | | | | |
| RCCA | 0.8971 | 0.9616 | 1.4573 | 1.5973 | 3.6553 |
| PLS | 0.9512 | 1.0852 | 1.4562 | 1.6106 | 3.6568 |
| PCR | 0.9586 | 1.0675 | 1.4504 | 1.6101 | 3.6904 |
| SNR = 10 | | | | | |
| RCCA | 0.5719 | 0.6281 | 0.9184 | 1.0119 | 1.8694 |
| PLS | 0.5930 | 0.6964 | 0.9325 | 1.0239 | 1.8733 |
| PCR | 0.6019 | 0.6823 | 0.9211 | 1.0240 | 1.8876 |
| SNR = 20 | | | | | |
| RCCA | 0.3816 | 0.4100 | 0.5676 | 0.6497 | 0.9395 |
| PLS | 0.3928 | 0.4507 | 0.5994 | 0.6733 | 0.9423 |
| PCR | 0.3946 | 0.4443 | 0.5872 | 0.6670 | 0.9508 |

TABLE 2: Steady state operating conditions of the distillation column.

| Process variable | Value | Process variable | Value |
|------------------|-------------------------|------------------|----------------------------|
| Feed | | | |
| F | 1 kg mole/sec | P | 1.7022×10^6 Pa |
| T | 322 K | x_D | 0.979 |
| P | 1.7225×10^6 Pa | Reboiler drum | |
| z_F | 0.4 | B | 0.5979 kg mole/sec |
| Reflux drum | | | |
| D | 0.40206 kg mole/sec | Q | 2.7385×10^7 Watts |
| T | 325 K | T | 366 K |
| Reflux | 62.6602 kg/sec | P | 1.72362×10^6 Pa |
| | | x_B | 0.01 |

TABLE 3: Comparison of the Monte Carlo MSE's for x_D in the simulated distillation column example.

| Model type | IMSLVR | MSF+LVR | EWMA+LVR | MF+LVR | LVR |
|------------------|--------|---------|----------|--------|--------|
| $\times 10^{-4}$ | | | | | |
| SNR = 5 | | | | | |
| RCCA | 0.0197 | 0.0205 | 0.0274 | 0.0286 | 0.0987 |
| PLS | 0.0202 | 0.0210 | 0.0288 | 0.0303 | 0.0984 |
| PCR | 0.0204 | 0.0212 | 0.0288 | 0.0357 | 0.0983 |
| $\times 10^{-5}$ | | | | | |
| SNR = 10 | | | | | |
| RCCA | 0.1279 | 0.1280 | 0.1700 | 0.1792 | 0.5403 |
| PLS | 0.1340 | 0.1341 | 0.1790 | 0.1891 | 0.5388 |
| PCR | 0.1317 | 0.1316 | 0.1778 | 0.1879 | 0.5423 |
| $\times 10^{-5}$ | | | | | |
| SNR = 20 | | | | | |
| RCCA | 0.0785 | 0.0791 | 0.1071 | 0.1157 | 0.3012 |
| PLS | 0.0844 | 0.0849 | 0.1130 | 0.1218 | 0.3017 |
| PCR | 0.0801 | 0.0803 | 0.1112 | 0.1200 | 0.3040 |

Tables 3 and 4 show that modeling prefiltered data significantly improves the prediction accuracy of the estimated LVR models over the conventional model estimation methods. The IMSLVR algorithm, however, improves the prediction of the estimated LVR model even further, especially at higher noise contents, that is, at smaller SNR. To illustrate the relative performances of the various LVR modeling techniques, as an example, the performances of the estimated RCCA models

for the top composition (x_D) in the case of SNR = 10 are shown in Figure 7.

6.3. Example 3: Dynamic LVR Modeling of an Experimental Packed Bed Distillation Column. In this example, the developed IMSLVR modeling algorithm is used to model a practical packed bed distillation column with a recycle

TABLE 4: Comparison of the Monte Carlo MSE's for x_B in the simulated distillation column example.

| Model type | IMSLVR | MSF+LVR | EWMA+LVR | MF+LVR | LVR |
|------------------|--------|---------|----------|--------|--------|
| $\times 10^{-5}$ | | | SNR = 5 | | |
| RCCA | 0.0308 | 0.0375 | 0.0685 | 0.0710 | 0.1972 |
| PLS | 0.0331 | 0.0393 | 0.0702 | 0.0725 | 0.1979 |
| PCR | 0.0327 | 0.0398 | 0.0708 | 0.0736 | 0.1961 |
| $\times 10^{-5}$ | | | SNR = 10 | | |
| RCCA | 0.0197 | 0.0206 | 0.0428 | 0.0447 | 0.1061 |
| PLS | 0.0212 | 0.0223 | 0.0448 | 0.0468 | 0.1063 |
| PCR | 0.0207 | 0.0214 | 0.0444 | 0.0466 | 0.1063 |
| $\times 10^{-6}$ | | | SNR = 20 | | |
| RCCA | 0.1126 | 0.1127 | 0.2623 | 0.2783 | 0.5653 |
| PLS | 0.1224 | 0.1222 | 0.2785 | 0.2956 | 0.5676 |
| PCR | 0.1183 | 0.1186 | 0.2736 | 0.2914 | 0.5703 |

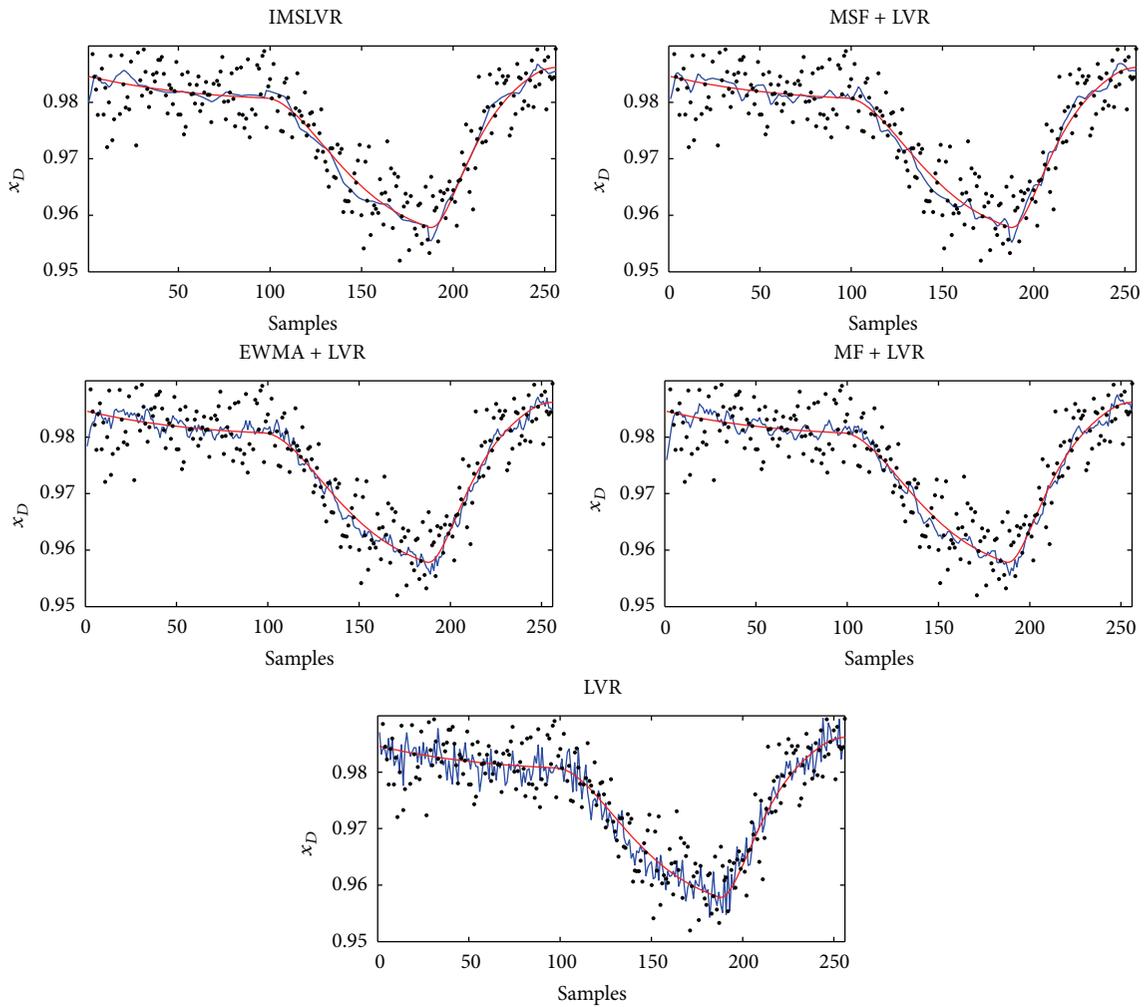
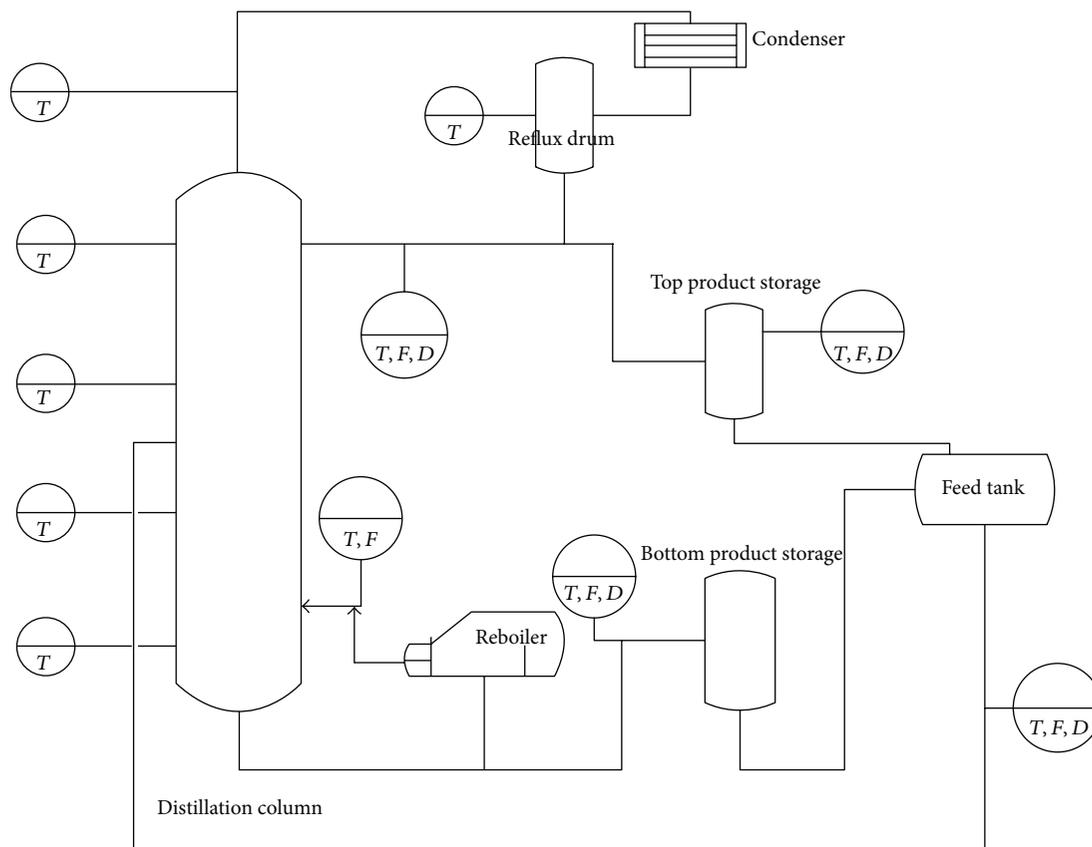


FIGURE 7: Comparison of the RCCA model predictions of x_D using the various LVR (RCCA) modeling techniques for the simulated distillation column example and the case where the noise SNR = 10 (solid blue line: model prediction; black dots: noisy data; solid red line: noise-free data).



T : temperature measurement sensor

F : flow measurement sensor

D : density measurement sensor

FIGURE 8: A schematic diagram of the packed bed distillation column setup.

TABLE 5: Steady state operating conditions of the packed bed distillation column.

| Process variable | Value |
|------------------|-------------------|
| Feed flow rate | 40 kg/hr |
| Reflux flow rate | 5 kg/hr |
| Feed composition | 0.3 mole fraction |
| Bottom level | 400 mm |

stream. More details about the process, data collection, and model estimation are presented next.

6.3.1. Description of the Packed Bed Distillation Column. The packed bed distillation column used in this experimental modeling example is a 6-inch diameter stainless steel column consisting of three packing sections (bottom, middle, and top section) rising to a height of 20 feet. The column, which is used to separate a methanol-water mixture, has Koch-Sulzer structured packing with liquid distributors above each packing section. An industrial quality Distributed Control System (DCS) is used to control the column. A schematic diagram

of packed bed distillation column is shown in Figure 8. Ten Resistance Temperature Detector (RTD) sensors are fixed at various locations in the setup to monitor the column temperature profile. The flow rates and densities of various streams (e.g., feed, reflux, top product, and bottom product) are also monitored. In addition, the setup includes four pumps and five heat exchangers at different locations.

The feed stream enters the column near its midpoint. The part of the column above the feed constitutes the rectifying section, and the part below (and including) the feed constitutes the stripping section. The feed flows down the stripping section into the bottom of the column, where a certain level of liquid is maintained by a closed-loop controller. A steam-heated reboiler is used to heat and vaporize part of the bottom stream, which is then sent back to the column. The vapor passes up the entire column contacting descending liquid on its way down. The bottom product is withdrawn from the bottom of the column and is then sent to a heat exchanger, where it is used to heat the feed stream. The vapors rising through the rectifying section are completely condensed in the condenser and the condensate is collected in the reflux drum, in which a specified liquid level is maintained.

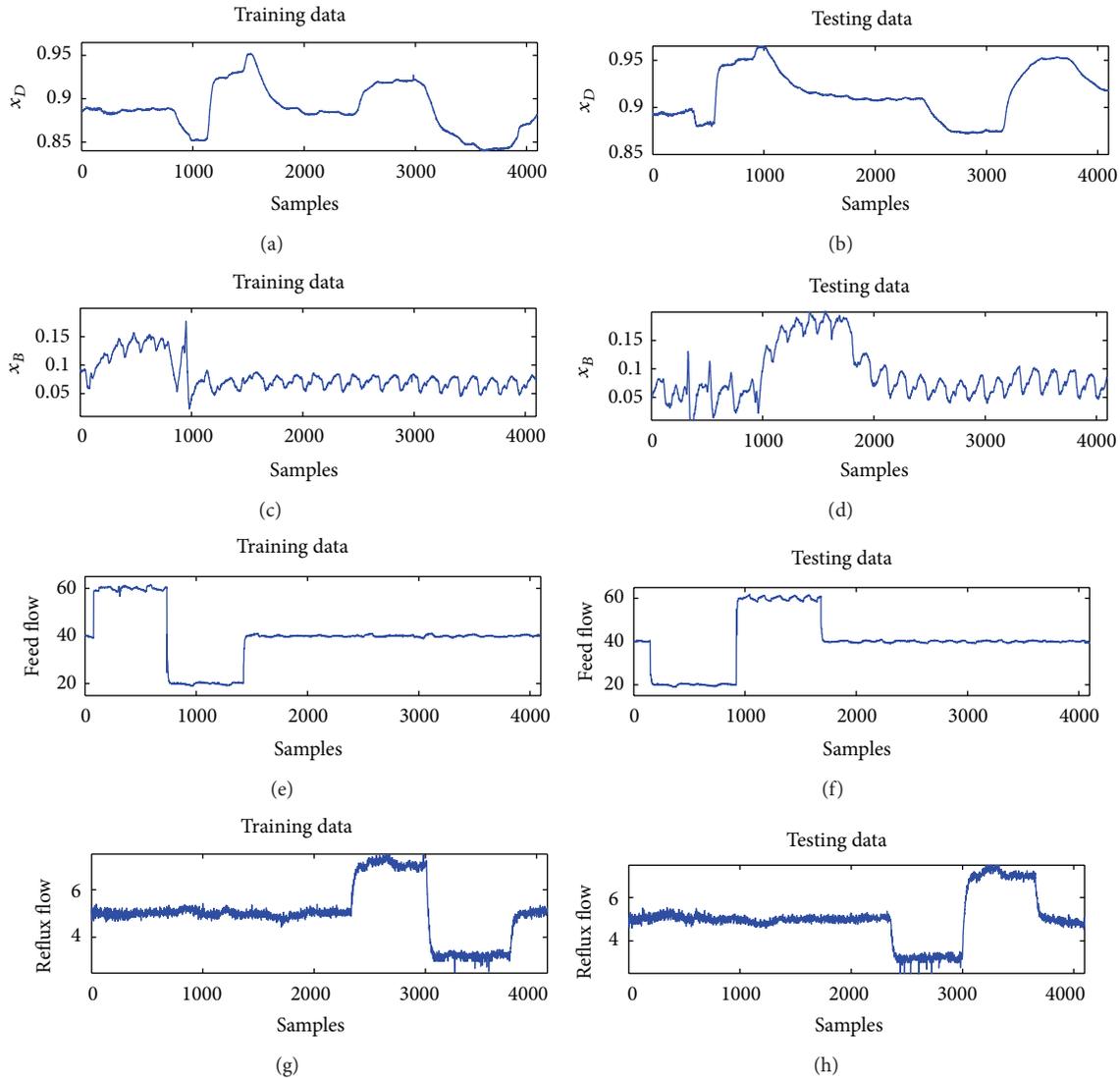


FIGURE 9: Training and testing data used in the packed bed distillation column modeling example.

A part of the condensate is sent back to the column using a reflux pump. The distillate not used as a reflux is cooled in a heat exchanger. The cooled distillate and bottom streams are collected in a feed tank, where they are mixed and later sent as a feed to the column.

6.3.2. Data Generation and Inferential Modeling. A sampling time of 4 s is chosen to collect the data used in this modeling problem. The data are generated by perturbing the flow rates of the feed and the reflux streams from their nominal operating values, which are shown in Table 5. First, step changes of magnitudes $\pm 50\%$ in the feed flow rate around its nominal value are introduced, and in each case, the process is allowed to settle to a new steady state. After attaining the nominal conditions again, similar step changes of magnitudes $\pm 40\%$ in the reflux flow rate around its nominal value are introduced. These perturbations are used to generate training and testing data (each consisting of 4096 data samples) to be

used in developing the various models. These perturbations are shown in Figures 9(e), 9(f), 9(g), and 9(h), and the effect of these perturbations on the distillate and bottom stream compositions are shown in Figures 9(a), 9(b), 9(c), and 9(d).

In this modeling problem, the input variables consist of six temperatures at different positions in the column, in addition to the flow rates of the feed and reflux streams. The output variables, on the other hand, are the compositions of the light component (methane) in the distillate and bottom streams (x_D and x_B , resp.). Because of the dynamic nature of the column and the presence of a recycle stream, the column always runs under transient conditions. These process dynamics can be accounted for in inferential models by including lagged inputs and outputs into the model [13, 45–48]. Therefore, in this dynamic modeling problem, lagged inputs and outputs are used in the LVR models to account for the dynamic behavior of the column. Thus, the model input matrix consists of 17 columns: eight columns for the inputs (the six temperatures and the flow rates of the feed

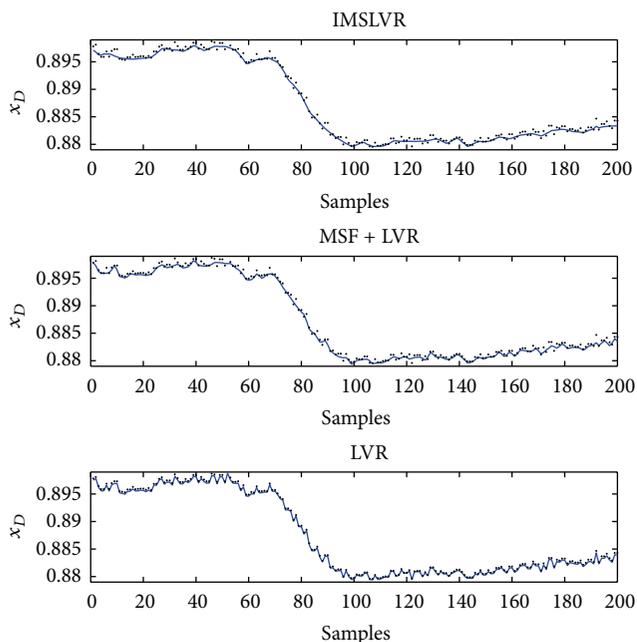


FIGURE 10: Comparison of the model predictions using the various modeling methods for the experimental packed bed distillation column example (solid blue line: model prediction; black dots: plant data).

and reflux streams), eight columns for the lagged inputs, and one column for the lagged output. To show the advantage of the IMSLVR algorithm, its performance is compared to those of the conventional LVR models and the models estimated using multiscale prefiltered data, and the results are shown in Figure 10. The results clearly show that multiscale prefiltering provides a significant improvement over the conventional LVR (RCCA) method (which sought to overfit the measurements), and that the IMSLVR algorithm provides further improvement in the smoothness and the prediction accuracy. Note that Figure 10 shows only a part of the testing data for the sake of clarity.

7. Conclusions

Latent variable regression models are commonly used in practice to estimate variables which are difficult to measure from other easier-to-measure variables. This paper presents a modeling technique to improve the prediction ability of LVR models by integrating multiscale filtering and LVR model estimation, which is called integrated multiscale LVR (IMSLVR) modeling. The idea behind the developed IMSLVR algorithm is to filter the input and output data at different scales, construct different models using the filtered data from each scale, and then select the model that provides the minimum cross-validation MSE. The performance of the IMSLVR modeling algorithm is compared to the conventional LVR modeling methods as well as modeling prefiltered data, either using low pass filtering (such as mean filtering or EMWA filtering) or using multiscale filtering through three examples, two simulated examples and one practical example.

The simulated examples use synthetic data and simulated distillation column data, while the practical example uses experimental packed bed distillation column data. The results of all examples show that data prefiltering (especially using multiscale filtering) provides a significant improvement over the conventional LVR methods, and that the IMSLVR algorithm provides a further improvement, especially at higher noise levels. The main reason for the advantages of the IMSLVR algorithm over modeling prefiltered data is that it integrates multiscale filtering and LVR modeling, which helps retain the model-relevant features in the data that can provide enhanced model predictions.

Acknowledgment

This work was supported by the Qatar National Research Fund (a member of the Qatar Foundation) under Grant NPRP 09-530-2-199.

References

- [1] B. R. Kowalski and M. B. Seasholtz, "Recent developments in multivariate calibration," *Journal of Chemometrics*, vol. 5, no. 3, pp. 129–145, 1991.
- [2] I. Frank and J. Friedman, "A statistical view of some chemometric regression tools," *Technometrics*, vol. 35, no. 2, pp. 109–148, 1993.
- [3] M. Stone and R. J. Brooks, "Continuum regression: cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression," *Journal of the Royal Statistical Society. Series B*, vol. 52, no. 2, pp. 237–269, 1990.
- [4] S. Wold, *Soft Modeling: The Basic Design and Some Extensions, Systems under Indirect Observations*, Elsevier, Amsterdam, The Netherlands, 1982.
- [5] E. C. Malthouse, A. C. Tamhane, and R. S. H. Mah, "Nonlinear partial least squares," *Computers and Chemical Engineering*, vol. 21, no. 8, pp. 875–890, 1997.
- [6] H. Hotelling, "Relations between two sets of variables," *Biometrika*, vol. 28, pp. 321–377, 1936.
- [7] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research*, vol. 3, no. 1, pp. 1–48, 2003.
- [8] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: an overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.
- [9] M. Borga, T. Landelius, and H. Knutsson, "A unified approach to pca, pls, mlr and cca, technical report," Tech. Rep., Linköping University, 1997.
- [10] J. V. Kresta, T. E. Marlin, and J. F. McGregor, "development of inferential process models using pls," *Computers & Chemical Engineering*, vol. 18, pp. 597–611, 1994.
- [11] T. Mejdell and S. Skogestad, "Estimation of distillation compositions from multiple temperature measurements using partial-least squares regression," *Industrial & Engineering Chemistry Research*, vol. 30, pp. 2543–2555, 1991.
- [12] M. Kano, K. Miyazaki, S. Hasebe, and I. Hashimoto, "Inferential control system of distillation compositions using dynamic

- partial least squares regression,” *Journal of Process Control*, vol. 10, no. 2, pp. 157–166, 2000.
- [13] T. Mejdell and S. Skogestad, “Composition estimator in a pilot-plant distillation column,” *Industrial & Engineering Chemistry Research*, vol. 30, pp. 2555–2564, 1991.
- [14] H. Yamamoto, H. Yamaji, E. Fukusaki, H. Ohno, and H. Fukuda, “Canonical correlation analysis for multivariate regression and its application to metabolic fingerprinting,” *Biochemical Engineering Journal*, vol. 40, no. 2, pp. 199–204, 2008.
- [15] B. R. Bakshi and G. Stephanopoulos, “Representation of process trends-IV. Induction of real-time patterns from operating data for diagnosis and supervisory control,” *Computers and Chemical Engineering*, vol. 18, no. 4, pp. 303–332, 1994.
- [16] B. Bakshi, “Multiscale analysis and modeling using wavelets,” *Journal of Chemometrics*, vol. 13, no. 3, pp. 415–434, 1999.
- [17] S. Palavajhala, R. Motrad, and B. Joseph, “Process identification using discrete wavelet transform: design of pre-filters,” *AICHE Journal*, vol. 42, no. 3, pp. 777–790, 1996.
- [18] B. R. Bakshi, “Multiscale PCA with application to multivariate statistical process monitoring,” *AICHE Journal*, vol. 44, no. 7, pp. 1596–1610, 1998.
- [19] A. N. Robertson, K. C. Park, and K. F. Alvin, “Extraction of impulse response data via wavelet transform for structural system identification,” *Journal of Vibration and Acoustics*, vol. 120, no. 1, pp. 252–260, 1998.
- [20] M. Nikolaou and P. Vuthandam, “FIR model identification: parsimony through kernel compression with wavelets,” *AICHE Journal*, vol. 44, no. 1, pp. 141–150, 1998.
- [21] M. N. Nounou and H. N. Nounou, “Multiscale fuzzy system identification,” *Journal of Process Control*, vol. 15, no. 7, pp. 763–770, 2005.
- [22] M. S. Reis, “A multiscale empirical modeling framework for system identification,” *Journal of Process Control*, vol. 19, pp. 1546–1557, 2009.
- [23] M. Nounou, “Multiscale finite impulse response modeling,” *Engineering Applications of Artificial Intelligence*, vol. 19, pp. 289–304, 2006.
- [24] M. N. Nounou and H. N. Nounou, “Improving the prediction and parsimony of ARX models using multiscale estimation,” *Applied Soft Computing Journal*, vol. 7, no. 3, pp. 711–721, 2007.
- [25] M. N. Nounou and H. N. Nounou, “Multiscale latent variable regression,” *International Journal of Chemical Engineering*, vol. 2010, Article ID 935315, 5 pages, 2010.
- [26] M. N. Nounou and H. N. Nounou, “Reduced noise effect in nonlinear model estimation using multiscale representation,” *Modelling and Simulation in Engineering*, vol. 2010, Article ID 217305, 8 pages, 2010.
- [27] J. F. Carrier and G. Stephanopoulos, “Wavelet-Based Modulation in Control-Relevant Process Identification,” *AICHE Journal*, vol. 44, no. 2, pp. 341–360, 1998.
- [28] M. Madakyaru, M. Nounou, and H. Nounou, “Linear inferential modeling: theoretical perspectives, extensions, and comparative analysis,” *Intelligent Control and Automation*, vol. 3, pp. 376–389, 2012.
- [29] R. Rosipal and N. Kramer, “Overview and recent advances in partial least squares,” in *Subspace, Latent Structure and Feature Selection*, Lecture Notes in Computer Science, pp. 34–51, Springer, New York, NY, USA, 2006.
- [30] P. Geladi and B. R. Kowalski, “Partial least-squares regression: a tutorial,” *Analytica Chimica Acta*, vol. 185, no. C, pp. 1–17, 1986.
- [31] S. Wold, “Cross-validatory estimation of the number of components in factor and principal components models,” *Technometrics*, vol. 20, no. 4, p. 397, 1978.
- [32] R. D. Strum and D. E. Kirk, *First Principles of Discrete Systems and Digital Signal Processing*, Addison-Wesley, Reading, Mass, USA, 1989.
- [33] M. N. Nounou and B. R. Bakshi, “On-line multiscale filtering of random and gross errors without process models,” *AICHE Journal*, vol. 45, no. 5, pp. 1041–1058, 1999.
- [34] G. Strang, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Wellesley, Mass, USA, 1986.
- [35] G. Strang, “Wavelets and dilation equations: a brief introduction,” *SIAM Review*, vol. 31, no. 4, pp. 614–627, 1989.
- [36] I. Daubechies, “Orthonormal bases of compactly supported wavelets,” *Communications on Pure and Applied Mathematics*, vol. 41, no. 7, pp. 909–996, 1988.
- [37] S. G. Mallat, “Theory for multiresolution signal decomposition: the wavelet representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.
- [38] A. Cohen, I. Daubechies, and P. Vial, “Wavelets on the interval and fast wavelet transforms,” *Applied and Computational Harmonic Analysis*, vol. 1, no. 1, pp. 54–81, 1993.
- [39] D. Donoho and I. Johnstone, “Ideal de-noising in an orthonormal basis chosen from a library of bases,” Tech. Rep., Department of Statistics, Stanford University, 1994.
- [40] D. L. Donoho, I. M. Johnstone, G. Kerkycharian, and D. Picard, “Wavelet shrinkage: asymptopia?” *Journal of the Royal Statistical Society. Series B*, vol. 57, no. 2, pp. 301–369, 1995.
- [41] M. Nounou and B. R. Bakshi, “Multiscale methods for denoising and compression,” in *Wavelets in Analytical Chemistry*, B. Walczak, Ed., pp. 119–150, Elsevier, Amsterdam, The Netherlands, 2000.
- [42] D. L. Donoho and I. M. Johnstone, “Ideal spatial adaptation by wavelet shrinkage,” *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.
- [43] G. P. Nason, “Wavelet shrinkage using cross-validation,” *Journal of the Royal Statistical Society. Series B*, vol. 58, no. 2, pp. 463–479, 1996.
- [44] M. N. Nounou, “Dealing with collinearity in fir models using bayesian shrinkage,” *Industrial and Engineering Chemistry Research*, vol. 45, pp. 292–298, 2006.
- [45] N. L. Ricker, “The use of biased least-squares estimators for parameters in discrete-time pulse-response models,” *Industrial and Engineering Chemistry Research*, vol. 27, no. 2, pp. 343–350, 1988.
- [46] J. F. MacGregor and A. K. L. Wong, “Multivariate model identification and stochastic control of a chemical reactor,” *Technometrics*, vol. 22, no. 4, pp. 453–464, 1980.
- [47] T. Mejdell and S. Skogestad, “Estimation of distillation compositions from multiple temperature measurements using partial-least-squares regression,” *Industrial & Engineering Chemistry Research*, vol. 30, no. 12, pp. 2543–2555, 1991.
- [48] T. Mejdell and S. Skogestad, “Output estimation using multiple secondary measurements: high-purity distillation,” *AICHE Journal*, vol. 39, no. 10, pp. 1641–1653, 1993.

