Probabilistic Slow Feature Analysis-Based Representation Learning from Massive Process Data for Soft Sensor Modeling

Chao Shang
Tsinghua National Laboratory for Information Science and Technology (TNList) and Dept. of Automation, Tsinghua University, Beijing 100084, P.R. China

Dept. of Chemical and Materials Engineering, University of Alberta, Edmonton, Alberta T6G 2G6, Canada

Biao Huang
Dept. of Chemical and Materials Engineering, University of Alberta, Edmonton, Alberta T6G 2G6, Canada

Fan Yang and Dexian Huang
Tsinghua National Laboratory for Information Science and Technology (TNList) and Dept. of Automation, Tsinghua University, Beijing 100084, P.R. China

DOI 10.1002/aic.14937
Published online July 18, 2015 in Wiley Online Library (wileyonlinelibrary.com)

Latent variable (LV) models provide explicit representations of underlying driving forces of process variations and retain the dominant information of process data. In this study, slow features (SFs) as temporally correlated LVs are derived using probabilistic SF analysis. SFs evolving in a state-space form effectively represent nominal variations of processes, some of which are potentially correlated to quality variables and hence help improving the prediction performance of soft sensors. An efficient expectation maximum algorithm is proposed to estimate parameters of the probabilistic model, which turns out to be suitable for analyzing massive process data. Two criteria are also proposed to select quality-relevant SFs. The validity and advantages of the proposed method are demonstrated via two case studies.
 AIChE J, 61: 4126–4139, 2015
Keywords: latent variable model, slow feature analysis, process data analysis, soft sensor

Introduction

Data-driven soft sensors have played an indispensable role in the process industries. As alternatives to hardware sensors, they provide real-time information about hard-to-measure variables based on online process data to facilitate control as well as monitoring of processes. In contrast to first-principle models, they have the capability to extract useful information from process data without resort to specific domain knowledge that is complicated in industrial processes and sometimes even unaccessible. Therefore, they have gained increasing popularity in a wide spectrum of industrial processes.[1,2]

Process data are characterized by their significant correlations. Strictly speaking, data space spanned by nominal operation data is of a reduced statistical rank. From a causal point of view, most variances in process data arise from inherent "common causes" such as raw materials fluctuations, environmental changes, and nominal disturbances.[3] This applies particularly to the cases with feedback control. Figure 1 depicts a typical diagram for chemical processes with feedback control. System disturbance not only affects the measured variables but also influences the manipulated variables in an indirect way, as the control system would adjust the manipulated variables according to deviations of quality indices, or measured variables in some occasions, from their setpoints. In this vein, latent variable (LV) models represent such "common causes" as low-dimensional LVs, which disentangle strong correlations between process data.[3] For soft sensing applications, the most-used LV models include principal component regression (PCR)[4,5] and partial least squares (PLS),[6,7] as well as their various extensions.

In recent years, a new paradigm named big data analytics has emerged in the machine learning area, and representation learning provides a new solution to model structure design as well as parameter training problems in the presence of big data.[8] For supervised learning tasks like classification and regression, classical models, such as support vector machines and neural networks, can directly establish the mapping between input X and output Y. In contrast, representation learning emphasizes particularly on the importance of first deriving a good representation of input X through unsupervised learning, which is also helpful for establishing a supervised predictor. Therefore, LV models can be seen as a good example of representation learning for process data analysis.[9]

Despite the prevalence of LV models, big data analytics provides more room to devise powerful representation of process data, as pointed out in the recent prospective article of Qin.[9] Because process data are measured with uniform

Correspondence concerning this article should be addressed to B. Huang at biao.huang@ualberta.ca.

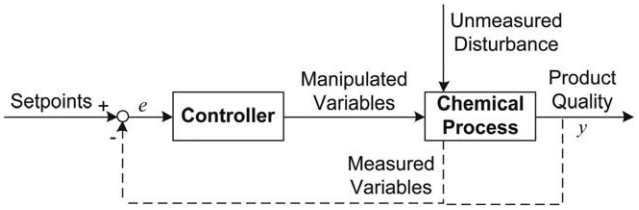

Figure 1. A systematic diagram for chemical processes with feedback control.

sampling periods, their dynamic nature conveys abundant information, in which temporal correlation is an important asset worthy of in-depth investigation. Temporal correlation, or auto-correlation, indicates that sequential time series samples tend to manifest some similarities. For process data representation, its notion could be further illustrated from the following two aspects:

- A LV as an appropriate representation, denoted as $s(t)$, should be associated with not only the current observations $\mathbf{x}(t)$ but also the preceding ones $\{\mathbf{x}(t-1), \mathbf{x}(t-2), \cdots\}$.
- $s(t)$ should be a causal system and correlated to its own preceding values $\{s(t-1), s(t-2), \cdots\}$.

These two issues appear to be similar but substantially differ from each other. For example, the former has already been addressed in classical LV models with lags, such as dynamic PCA (DPCA)[10] and dynamic PLS (DPLS),[11] in which lagged observations in a period of time are used for inferring LVs. The latter essentially indicates that, LVs as the underlying driving causes of chemical processes be auto-correlated a priori, or in other words, slowly varying. For a stationary and ergodic signal, its slowly varying nature can be perceived as that the power spectral density $S_x(\omega)$ is concentrated in the low frequency, which is associated with a strong auto-correlation $R_x(\tau)$ thereof according to $S_x(\omega) = \sum_{-\infty}^{+\infty} R_x(\tau)e^{-j\omega\tau}d\tau$. It is an intuitive truth that, the LV $s(t)$, on behalf of the most essential part of processes, should have a significant temporal correlation rather than abrupt fluctuations because of inertia characteristics. However, such information is unfortunately ignored by classical models, since LV samples in DPCA and DPLS in different time slices are still assumed to be independent without any specific priors being imposed on.

In order to further improve model interpretations, slow feature analysis (SFA), a novel machine learning approach proposed by Wiskott and Sejnowski,[12] has been exploited to learn temporally correlated representations for process monitoring,[13] as well as quality prediction in a preliminary study.[14] SFA adopts the temporal coherence as a heuristic prior to induce meaningful LVs, referred to as slow features (SFs). If the input signals are driven by some inherent trends with significant auto-correlations, SFA could always extract mutually uncorrelated LVs with different varying frequencies. Moreover, its probabilistic extension provides a more compact and clearer description to process dynamics with a state-space form, as to be conducted in this work to approach the soft sensing problem from the viewpoint of representation learning. Probabilistic SFA (PSFA) is first applied to fast-rate process data to derive meaningful SFs, some of which are further chosen as quality predictors. The rationale of the proposed method lies particularly in that, in routine operations with desirable feedback control, the process is subject to nominal "common cause" variations, a part of which may exert influence on quality variables intrinsically; therefore, SFs can disentangle such nominal variations with discrepant frequencies. It then enables some quality-relevant SFs to be clearly represented and further readily selected, which are beneficial for final quality predictions. We apply the expectation maximum (EM) algorithm to the parameter estimation problem of PSFA. To improve the optimization performance of the EM algorithm, the relationships between PSFA, linear SFA and dynamic SFA (DSFA) are exploited and analyzed, based on which a novel initialization strategy is established to enhance the performance of the EM algorithm. It endows development of the PSFA model with a low computational cost and the ease of implementation in the presence of massive process data. Two simple criteria, which are based on slowness and correlation respectively, are also proposed to select the potential quality-relevant SFs as predictors of soft sensors. Compared with generic soft sensor models like DPLS and dynamic PCR (DPCR), the proposed method is able to accommodate various process dynamics with a compact structure, and allows an unequal number of input and output samples such that both fast-rate process data and slow-rate quality data could be appropriately synthesized. An experimental example on a hybrid tank system is used to clarify the basics of the proposed method. Finally, empirical results on an industrial dataset demonstrate the validity of SFs and the improvement of prediction accuracy.

The remainder of this article proceeds as follows. The next section gives an overview of SFA. Its probabilistic interpretation, that is, PSFA, is motivated and detailed in "Probabilistic SF Analysis" in which the parameter estimation method based on the EM algorithm as well as an efficient initialization strategy is also developed. In "Soft Sensor Modeling based on SFs" the proposed soft sensor modeling scheme is introduced, and some practical merits are discussed. In "Experimental Case Study: Hybrid Tank System" and "Industrial Case Study: SRU Process," an experimental example as well as an industrial case is utilized to demonstrate the validity of the proposed approach. Finally, concluding remarks are presented.

Slow Feature Analysis—Revisit

The theoretical framework of SFA algorithm was first established in 2002 by Wiskott and Sejnowski.[12] It is interesting to mention that the first application of SFA appeared in the computational neuroscience area, with the aim to help understand the organization of the visual system in the brain.[15,16] Later, SFA has been successfully applied in multifarious learning tasks, such as object recognition,[17] change detection,[18,19] and nonlinear blind source separation.[20–22] In this section, the basics of classical SFA and its usage in unsupervised dimension reduction are reviewed. First, we clarify some notations and definitions that are useful in the sequel, and then detail the mathematical formulation of SFA along with the associated optimization problem.

Definition of slowness and some notations

Given a stochastic and ergodic signal $X(t)$, how slow or how fast $X(t)$ varies, is quantitatively measured as

$$\Delta(X(t)) \triangleq \langle \dot{X}^2(t) \rangle_t \tag{1}$$

wherein the operator $\langle \cdot \rangle_t$ denotes the expectation empirically calculated by time averaging

$$\mathbb{E}\{X(t)\} \approx \langle X(t)\rangle_t \triangleq \frac{1}{t_1-t_0}\int_{t_0}^{t_1} X(t)dt \tag{2}$$

and $\dot{X}(t)$ stands for the derivative of $X(t)$ with respect to time. The symbol $\Delta(\cdot)$ can be viewed a natural definition of slowness of a certain signal.

In industrial application scenarios, process data $X(t)$ are measured based on a discrete sampling interval. Assume that we have a section of time series samples denoted as $\{X(1), X(2), \cdots, X(T)\}$, where T is the number of archived samples. Therefore, the time averaging for deriving the expectation operator in (2) could be estimated with discrete samples

$$\langle X(t)\rangle_t \triangleq \frac{1}{t_1-t_0}\int_{t_0}^{t_1} X(t)dt \approx \frac{1}{T}\sum_{t=1}^{T} X(t) \tag{3}$$

Likewise, the temporal difference could be used to replace the temporal derivative as an approximation

$$\dot{X}(t) = \frac{dX(t)}{dt} \approx X(t)-X(t-1) \tag{4}$$

Linear SFA

Assume that an input vector $\mathbf{x}(t)$ is mapped to a q-dimensional feature space $\mathcal{F}$ by a set of real-valued functions $\{g_1(\mathbf{x}(t)), \ldots, g_q(\mathbf{x}(t))\}$ such that the LVs of primary interest, which are named as SFs, are expressed as the outputs of these real-valued functions $s_j(t) \triangleq g_j(\mathbf{x}(t))(1 \le j \le q)$. The objective of SFA is then to make the variations of SFs as slow as possible, which is identical to finding a set of real-valued functions $\{g_j(\cdot),\ 1 \le j \le q\}$ that minimize

$$\min_{g_j(\cdot)} \Delta(s_j) \tag{5}$$

under the constraints

$$\langle s_j(t)\rangle_t = 0,\ \text{(zero mean)} \tag{6}$$

$$\langle s_j^2(t)\rangle_t = 1,\ \text{(unit variance)} \tag{7}$$

$$\forall i \ne j,\ \langle s_i(t)s_j(t)\rangle_t = 0.\ \text{(decorrelation and order)} \tag{8}$$

It is obvious that any constant-valued function, $g_j(\mathbf{x}) \equiv \text{const}$, would yield an optimal solution if only objective (1) is taken into account. Therefore, Constraints (6) and (7) are supplemented to scale SFs $\{s_j\}$ to zero mean and unit variance with the intent to avoid trivial solutions. Constraint (8) enforces outputs of functions $\{g_j\}$ to be mutually independent, resulting in a natural descending order of $\{s_j\}$ in which the most slowly varying signal has the lowest index. For example, the SF $s_1(t)$ denotes the slowest one, while $s_2(t)$ denotes the second slowest one, which also keeps uncorrelated to $s_1(t)$.

The linear SFA algorithm

Linear SFA assumes that SFs $\mathbf{s}(t)$ are derived with linear mappings from input $\mathbf{x}(t)$, that is

$$\mathbf{s}(t) = \mathbf{W}^{\mathrm{T}}\mathbf{x}(t) \tag{9}$$

where $\mathbf{W} = \begin{bmatrix}\mathbf{w}_1\, \mathbf{w}_2\, \cdots \mathbf{w}_q\end{bmatrix} \in \mathbb{R}^{m\times q}$ is the coefficient matrix to be optimized. To enforce satisfaction of the zero mean constraint (6), data along each dimension of inputs $\mathbf{x}(t)$ are assumed to have zero mean. When the number of SFs is equal to that of model inputs, i.e. $q = m$, the optimization problem in (1) can be readily solved with the Lagrange multiplier, yielding the following generalized eigenvalue problem[12]

$$\mathbf{AW} = \mathbf{BW\Omega} \tag{10}$$

where $\mathbf{A} = \langle \dot{\mathbf{x}}\dot{\mathbf{x}}^{\mathrm{T}}\rangle_t$ is the covariance matrix of the temporal derivatives of input $\mathbf{x}$ with entries

$$A_{ij} = \langle \dot{x}_i(t)\dot{x}_j(t)\rangle_t \tag{11}$$

and $\mathbf{B} = \langle \mathbf{x}\mathbf{x}^{\mathrm{T}}\rangle_t$ is the covariance matrix of $\mathbf{x}$ with entries

$$B_{ij} = \langle x_i(t)x_j(t)\rangle_t \tag{12}$$

Matrix Ω is a diagonal matrix that contains the generalized eigenvalues $\{\omega_j\}$ on its diagonal, which are exactly the optimal values of objectives in (1)

$$\Delta(s_j) = \langle \dot{s}_j^2(t)\rangle_t = \omega_j \tag{13}$$

Dynamic SFA

Notice that in the conventional SFA, feature functions $\{g_j(\mathbf{x}(t))\}$ are computed using process data at the current snapshot t. Consequently, SFs cannot be achieved by filtering time series historical data, in the sense that LVs are irrelevant to its earlier historical samples. In industrial processes, however, latent states ought to be relevant with historical process data over a certain period of time. To further extract dynamic information in time series data, conventional linear SFA can be extended to a dynamic version (DSFA) by appending d lagged measurements

$$\tilde{\mathbf{x}}(t) \triangleq \begin{bmatrix}\mathbf{x}(t)\\ \mathbf{x}(t-1)\\ \vdots\\ \mathbf{x}(t-d)\end{bmatrix} \in \mathbb{R}^{m(d+1)} \tag{14}$$

as in the case of DPCA[10] and DPLS.[11]

Probabilistic SF Analysis

Many classical LV models have been translated into probabilistic graphical models (PGMs), such as probabilistic PCA[23] and probabilistic independent component analysis.[24] In virtue of probabilistic interpretations, PGMs not only provide better insights into the model properties but also become essentially generative models that permit drawing new data samples from a given probability distribution. In addition, Bayesian inference and learning strategies have already been systematically established in a probabilistic framework,[25] which allow easy extensions to newly devised PGMs. In this section, we introduce PSFA, a conceptually simple state-space model with a Markov chain architecture, which inherits the associated advantages of PGMs. We make use of the fact that PSFA is identical to the linear SFA in the limiting case, and design ad hoc parameter learning methods for PSFA.

Mathematical formulation

Mathematically, PSFA is formulated as[26]

$$\begin{aligned} &s_j(t) = \lambda_j s_j(t-1)+e_j(t),\ e_j(t) \sim \mathcal{N}(0, 1-\lambda_j^2),\ 1 \le j \le q\\ &\mathbf{x}(t) = \mathbf{H}\mathbf{s}(t)+\mathbf{e}_x,\ \mathbf{e}_x \sim \mathcal{N}(\mathbf{0}, \Sigma),\ \Sigma = \mathrm{diag}\{\sigma_1^2, \cdots, \sigma_m^2\}. \end{aligned} \tag{15}$$

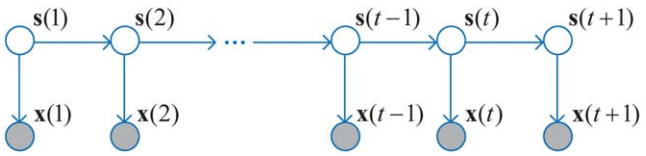

Figure 2. Graphical structure of Probabilistic SFA; grey nodes represent observed variables.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

In a probabilistic setting, SFs $\{s_j(t)\}$ are modeled as a series of independent auto-regressive AR (1) processes, each of which is consistently corrupted by an independent noise $e_j(t)$ following Gaussian distribution. The decorrelation nature in Constraint (8) is characterized by the independence assumption of $\{s_j(t)\}$. One can easily verify that the stationary distribution has zero mean and unit variance

$$\mathbb{E}\{s_j(t)\} = 0,\ \mathrm{Var}\{s_j(t)\} = 1,\ 1 \le j \le q \tag{16}$$

which are in line with Constraints (6) and (7) of linear SFA. The slowly varying nature of $\{s_j(t)\}$ arises from its Markov property. The correlation level between adjacent data points $s_j(t)$ and $s_j(t-1)$ is governed by the transition parameter λ_j, which satisfies $0 \le \lambda_j < 1$. In fact, its slowness measure $\Delta(\cdot)$ can be conveniently calculated as

$$\begin{aligned}
&\Delta(s_j) = \omega_j \\
=\ &\langle (s_j(t)-s_j(t-1))^2 \rangle_t \\
=\ &\langle (\lambda_j s_j(t-1)+e_j(t)-s_j(t-1))^2 \rangle_t \\
=\ &(1-\lambda_j)^2 \langle s_j^2(t-1)\rangle_t + \langle e_j^2(t)\rangle_t \\
=\ &(1-\lambda_j)^2+1-\lambda_j^2 \\
=\ &2(1-\lambda_j).
\end{aligned} \tag{17}$$

Intuitively, a large λ_j implies a strong correlation between $s_j(t)$ and $s_j(t-1)$, and indicates that $s_j(t)$ tends to have slow variations with a small $\Delta(\cdot)$ value, and vice versa. Apart from specific priors on SFs, a linear mapping from SFs $\mathbf{s}(t)$ to observations $\mathbf{x}(t)$ is assumed, plus a measurement noise term $\mathbf{e}_x(t)$, the covariance matrix of which is denoted as Σ. The graphical structure of PSFA is depicted in Figure 2, in which each observation $\mathbf{s}(t)$ of SFs is conditioned on its preceding observation $\mathbf{s}(t-1)$, and the process input $\mathbf{x}(t)$ is conditioned on the corresponding SFs $\mathbf{s}(t)$.

One could easily rewrite the formulation of PSFA in a general form of linear dynamical systems (LDS)[27,28]

$$\begin{aligned}
\mathbf{s}(t) &= \mathbf{F}\mathbf{s}(t-1)+\mathbf{e}(t) \\
\mathbf{x}(t) &= \mathbf{H}\mathbf{s}(t)+\mathbf{e}_x \\
\mathbf{e}(t) &\sim \mathcal{N}(\mathbf{0},\Gamma) \\
\mathbf{e}_x &\sim \mathcal{N}(\mathbf{0},\Sigma)
\end{aligned} \tag{18}$$

where $\mathbf{F}$ and Γ are diagonal matrices defined as:

$$\mathbf{F} = \mathrm{diag}\{\lambda_1,\cdots,\lambda_q\},\ \Gamma = \mathrm{diag}\left\{1-\lambda_1^2,\cdots,1-\lambda_q^2\right\}, \tag{19}$$

and $\Sigma = \mathrm{diag}\{\sigma_1^2,\cdots,\sigma_m^2\}$. In summary, the unknown parameters to be estimated are transition parameters $\{\lambda_j,\ 1 \le j \le q\}$, emission matrix $\mathbf{H} \in \mathbb{R}^{m\times q}$, and measurement noise variances $\{\sigma_i^2,\ 1 \le i \le m\}$. Notice that PSFA assigns no requirements to the relation between m and q. For linear SFA, the relationship $q \le m$ must hold. However, PSFA permits the case of over-complete features[29] with $q > m$.

As aforementioned in the Introduction section, LVs can be seen as abstractions of "common cause" disturbances or perturbations of processes. For PSFA, disturbances $\mathbf{s}(t)$ are modeled as a series of AR (1) Markov processes, while for PCA and PLS, disturbances are modeled as independent and identical Gaussian without temporal relations. In this sense, probabilistic SFR regards "common causes" as dynamics, which is entitled to more meaningful information than classical LV models. Therefore, the proposed model achieves an enhanced temporal description for sequential process data.

Formal Equivalence Between PSFA and Linear SFA. Assume that we have time series data $\{\mathbf{x}(1),\mathbf{x}(2),\cdots,\mathbf{x}(T)\}$ where T is the number of sequential samples. Turner and Sahani[26] have proved that, the maximum-likelihood estimation of PSFA will exactly recover the solution of linear SFA provided that $m = q$, $\lambda_i \ne \lambda_j\,(i \ne j)$, $T \to \infty$ and $\Sigma \to \mathbf{0}$. Readers are referred to Turner and Sahani's work for more mathematical details about the proof. A critical information conveyed by this equivalence is that, linear SFA is able to handle the noise-free case with $\{\sigma_i^2 = 0,\ 1 \le i \le m\}$ only; in the noisy case with $\{\sigma_i^2 > 0,\ 1 \le i \le m\}$, however, adopting the probabilistic formulation will be necessary for the denoising purpose. In addition, the solution of linear SFA can be regarded as an approximation to that of PSFA.

Parameter estimation using the EM algorithm

In general, parameter optimization of PGMs can be recast as a maximum-likelihood estimation problem to the incomplete data likelihood $P(\mathbf{X}|\theta)$, where parameters of PSFA are uniformly denoted as $\theta \triangleq \{\lambda_j, 1 \le j \le q, \mathbf{H}, \Sigma\}$. Unfortunately, direct optimization of $P(\mathbf{X}|\theta)$ with respect to θ is occasionally intractable. The EM algorithm provides an effective solution to the parameter estimation problem, which iterates between the expectation step (E-step) and the maximization step (M-step).[30,31] EM algorithm has been well founded for inferring parameters of LDSs.[27,28,32] Turner and Sahani have pointed out that the EM algorithm is applicable to the parameter estimation problem of PSFA thanks to a simplified LDS formulation[26]; however, no methodological details are given in their work. In this article, we give explicitly the EM algorithm for PSFA, which entails some modifications based on the existing work on LDS.[27,28,32] Given time series observations $\mathbf{X} = \{\mathbf{x}(t),\ 1 \le t \le T\}$, the complete data log-likelihood of a LDS can be written as[32]

$$\begin{aligned}
\log P(\mathbf{X},\mathbf{S}|\theta) &= \log P(\mathbf{s}(1)) \\
&+\sum_{t=2}^{T} \log P(\mathbf{s}(t)|\mathbf{s}(t-1)) + \sum_{t=1}^{T} \log P(\mathbf{x}(t)|\mathbf{s}(t)).
\end{aligned} \tag{20}$$

Because $\{s_j(t), 1 \le j \le q\}$ are stationary stochastic processes, the initial states $\mathbf{s}(1)$ are assumed to follow a Gaussian distribution $N(\mathbf{0},\mathbf{I}_q)$. Therefore (20) can be decomposed as

$$
\begin{aligned}
&\log P(\mathbf{X},\mathbf{S}|\theta)\\
&=-\frac{1}{2}\mathbf{s}(1)^{\mathrm{T}}\mathbf{s}(1)-\frac{T-1}{2}\sum_{j=1}^{q}\log(1-\lambda_j^2)\\
&-\frac{1}{2}\sum_{t=2}^{T}\sum_{j=1}^{q}\frac{1}{1-\lambda_j^2}\left[s_j(t)-\lambda_j s_j(t-1)\right]^2\\
&-\frac{T}{2}\log\det\Sigma-\frac{1}{2}\sum_{t=1}^{T}[\mathbf{x}(t)-\mathbf{H}\mathbf{s}(t)]^{\mathrm{T}}\Sigma^{-1}[\mathbf{x}(t)-\mathbf{H}\mathbf{s}(t)]\\
&-\frac{q(m+T)}{2}\log(2\pi)
\end{aligned}
\tag{21}
$$

Then the Q-function can be formally derived by considering the conditional expectation of (21)

$$
Q(\theta,\theta^{\mathrm{old}})=\mathbb{E}_{\mathbf{X},\theta^{\mathrm{old}}}\{\log P(\mathbf{X},\mathbf{S}|\theta)\} \tag{22}
$$

where θ^{old} denotes the parameters in the previous iteration.

M-Step. In the M-step, the Q-function is maximized with respect to θ to obtain

$$
\theta^{\mathrm{new}}=\arg\max_{\theta}Q(\theta,\theta^{\mathrm{old}}) \tag{23}
$$

Because both matrices $\mathbf{F}$ and Γ share the same parameters $\{\lambda_j\}$ in PSFA, the optimization of $\{\lambda_j\}$ in the M-step differs from that for general LDSs. Taking the derivative of Q-function with respect to λ_j, we derive

$$
\frac{\partial Q}{\partial\lambda_j}=\frac{T-1}{2}\cdot\frac{2\lambda_j}{1-\lambda_j^2}-\frac{1}{2}\cdot\frac{(1-\lambda_j^2)J'(\lambda_j)+2\lambda_j J(\lambda_j)}{(1-\lambda_j^2)^2} \tag{24}
$$

where

$$
\begin{aligned}
&J(\lambda_j)=\sum_{t=2}^{T}\mathbb{E}_{\mathbf{X},\theta^{\mathrm{old}}}\left\{\left[s_j(t)-\lambda_j s_j(t-1)\right]^2\right\},\\
&J'(\lambda_j)=2\lambda_j\sum_{t=2}^{T}\mathbb{E}_{\mathbf{X},\theta^{\mathrm{old}}}\left\{s_j^2(t-1)\right\}-2\sum_{t=2}^{T}\mathbb{E}_{\mathbf{X},\theta^{\mathrm{old}}}\left\{s_j(t)s_j(t-1)\right\},\ 1\le j\le q
\end{aligned}
\tag{25}
$$

Substituting (25) into (24) and setting (24) to zero yield the following cubic equation

$$
a_{j_3}\lambda_j^3+a_{j_2}\lambda_j^2+a_{j_1}\lambda_j+a_{j_0}=0 \tag{26}
$$

where the coefficients of (26) are derived as

$$
\begin{aligned}
a_{j_0}&=-\sum_{t=2}^{T}\mathbb{E}_{\mathbf{X},\theta^{\mathrm{old}}}\{s_j(t)s_j(t-1)\},\\
a_{j_1}&=\sum_{t=2}^{T}\left[\mathbb{E}_{\mathbf{X},\theta^{\mathrm{old}}}\{s_j^2(t)+s_j^2(t-1)\}-1\right],\\
a_{j_2}&=-\sum_{t=2}^{T}\mathbb{E}_{\mathbf{X},\theta^{\mathrm{old}}}\{s_j(t)s_j(t-1)\},\\
a_{j_3}&=T-1
\end{aligned}
\tag{27}
$$

Therefore, the updated λ_j^{new} could be calculated as the root of (26) within the range $[0,1)$. Similarly, by setting the partial derivatives of (21) to zero with respect to matrices $\mathbf{H}$ and $\{\sigma_i^2,\ 1\le i\le m\}$, the updated emission matrix and noise variances are given by[32]

$$
\begin{aligned}
\mathbf{H}^{\mathrm{new}}&=\left(\sum_{t=1}^{T}\mathbb{E}_{\mathbf{X},\theta^{\mathrm{old}}}\{\mathbf{x}(t)\mathbf{s}^{\mathrm{T}}(t)\}\right)\left(\sum_{t=1}^{T}\mathbb{E}_{\mathbf{X},\theta^{\mathrm{old}}}\{\mathbf{s}(t)\mathbf{s}^{\mathrm{T}}(t)\}\right)^{-1},\\
\left(\sigma_i^2\right)^{\mathrm{new}}&=\frac{1}{T}\sum_{t=1}^{T}\{\mathbb{E}_{\mathbf{X},\theta^{\mathrm{old}}}\{x_i^2(t)\}-2\left(\mathbf{h}_{i\cdot}^{\mathrm{T}}\right)^{\mathrm{new}}\mathbb{E}_{\mathbf{X},\theta^{\mathrm{old}}}\{\mathbf{s}(t)x_i(t)\}\\
&\quad+\left(\mathbf{h}_{i\cdot}^{\mathrm{T}}\right)^{\mathrm{new}}\mathbb{E}_{\mathbf{X},\theta^{\mathrm{old}}}\{\mathbf{s}(t)\mathbf{s}^{\mathrm{T}}(t)\}(\mathbf{h}_{i\cdot})^{\mathrm{new}}\},\ 1\le i\le m
\end{aligned}
\tag{28}
$$

where $\mathbf{h}_{i\cdot}^{\mathrm{T}}$ denotes the ith row of matrix $\mathbf{H}$.

E-Step. The above M-step involves evaluating expectations of the posterior distribution $P(\mathbf{S}|\mathbf{X},\theta^{\mathrm{old}})$[32]

$$
\begin{aligned}
\mathbb{E}_{\mathbf{X},\theta^{\mathrm{old}}}\{\mathbf{s}(t)\}&=\hat{\mu}_t,\\
\mathbb{E}_{\mathbf{X},\theta^{\mathrm{old}}}\{\mathbf{s}(t)\mathbf{s}^{\mathrm{T}}(t-1)\}&=\mathbf{J}_{t-1}\hat{\mathbf{V}}_t+\hat{\mu}_t\hat{\mu}_{t-1}^{\mathrm{T}},\\
\mathbb{E}_{\mathbf{X},\theta^{\mathrm{old}}}\{\mathbf{s}(t)\mathbf{s}^{\mathrm{T}}(t)\}&=\hat{\mathbf{V}}_t+\hat{\mu}_t\hat{\mu}_t^{\mathrm{T}}
\end{aligned}
\tag{29}
$$

which are to be calculated in the E-step. Because PSFA pertains to the general LDS, their E-steps are substantially equivalent, and thus the algorithm in the existing work[32] is directly given in this stage. The inference of latent states $\mathbf{S}$ given observations $\mathbf{X}$ is essentially tantamount to the Kalman smoothing problem,[33] the solution to which typically consists of the forward recursion and the backward recursion. First, forward recursions are used to calculate the posterior distribution $P(\mathbf{s}(t)|\mathbf{x}(1),\cdots,\mathbf{x}(t),\theta^{\mathrm{old}})\sim\mathcal{N}(\mu_t,\mathbf{P}_t)$ in a sequential manner[33]

$$
\begin{aligned}
\mathbf{P}_{t-1}&=\mathbf{F}\mathbf{V}_{t-1}\mathbf{F}^{\mathrm{T}}+\Gamma,\\
\mu_t&=\mathbf{F}\mu_{t-1}+\mathbf{K}_t[\mathbf{x}(t)-\mathbf{H}\mathbf{F}\mu_{t-1}],\\
\mathbf{V}_t&=(\mathbf{I}-\mathbf{K}_t\mathbf{H})\mathbf{P}_{t-1},\\
\mathbf{K}_t&=\mathbf{P}_{t-1}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}_{t-1}\mathbf{H}^{\mathrm{T}}+\Sigma)^{-1}
\end{aligned}
\tag{30}
$$

with initializations

$$
\begin{aligned}
\mu_1&=\mathbf{K}_1\mathbf{x}(1),\\
\mathbf{V}_1&=\mathbf{I}-\mathbf{K}_1\mathbf{H},\\
\mathbf{K}_1&=\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{H}^{\mathrm{T}}+\Sigma\right)^{-1}
\end{aligned}
\tag{31}
$$

Finally, parameters of the posterior distribution $P(\mathbf{S}|\mathbf{X},\theta^{\mathrm{old}})$ are obtained via backward recursions[33]

$$
\begin{aligned}
\hat{\mu}_t&=\mu_t+\mathbf{J}_t(\hat{\mu}_{t+1}-\mathbf{F}\mu_t),\\
\hat{\mathbf{V}}_t&=\mathbf{V}_t+\mathbf{J}_t(\hat{\mathbf{V}}_{t+1}-\mathbf{P}_t)\mathbf{J}_t^{\mathrm{T}},\\
\mathbf{J}_t&=\mathbf{V}_t\mathbf{F}^{\mathrm{T}}\mathbf{P}_t^{-1}
\end{aligned}
\tag{32}
$$

with initializations

$$
\hat{\mu}_T=\mu_T,\ \hat{\mathbf{V}}_T=\mathbf{V}_T \tag{33}
$$

Evaluation of the Incomplete Data Likelihood. In the EM algorithm derived above, the log-likelihood of complete data $\log P(\mathbf{X},\mathbf{S}|\theta)$ is available. For monitoring the convergence of EM algorithm as well as the model performance, however, the log-likelihood of observed data $\log P(\mathbf{X}|\theta)$ should be considered by integrating $\log P(\mathbf{X},\mathbf{S}|\theta)$ with respect to hidden variables $\mathbf{S}$, which is computationally involved. Fortunately, $\log P(\mathbf{X}|\theta)$ could be attained by using parameters derived in

the forward recursions, as introduced above. To see this, the conditional distributions over the observed data are first defined as

$$c_t \triangleq P(\mathbf{x}(t)|\mathbf{x}(1),\mathbf{x}(2),\cdots,\mathbf{x}(t-1)), 1 \le t \le T \tag{34}$$

which follows a Gaussian distribution[32]

$$c_t \sim \mathcal{N}(\mathbf{x}(t)|\mathbf{HF}\mu_{t-1}, \mathbf{HP}_{t-1}\mathbf{H}^{\mathrm{T}}+\Sigma) \tag{35}$$

According to the product rule, the incomplete data likelihood can be expressed as

$$P(\mathbf{X}|\theta) = P(\mathbf{x}(1),\mathbf{x}(2),\cdots,\mathbf{x}(t)|\theta) = \prod_{t=1}^{T} c_t \tag{36}$$

Then the log-likelihood of observed data can be calculated as

$$\log P(\mathbf{X}|\theta) = \log\left(\prod_{t=1}^{T} c_t\right) = \sum_{t=1}^{T} \log c_t \tag{37}$$

Improved Initialization with Linear SFA. The performance of the EM algorithm heavily depends on the initial solution. In this regard, the EM algorithm is generally carried out several times with random initializations, and the best solution is selected as the one with the largest likelihood value hereafter. However, for probabilistic models with a chain structure like PSFA, each run of the EM algorithm will be computationally formidable, not to mention several runs. It is therefore imperative to develop specific strategies to reduce the computational cost involved in training PSFA. Here we propose a useful initialization approach to improve the efficiency of the EM algorithm.

The equivalence between PSFA and the linear SFA implies that, the solution of linear SFA could be considered as an approximation to the maximum-likelihood estimation of PSFA. In terms of the slowness measure $\Delta(\cdot)$, the following relationship has been established

$$\Delta(s_j) = \omega_j = 2(1-\lambda_j) \tag{38}$$

Therefore, it is reasonable to use the solution of (10) to generate the initial guess of $\{\lambda_j\}$

$$\lambda_j = 1-\frac{\omega_j}{2} \tag{39}$$

For linear SFA, when $m = q$, the original input $\mathbf{x}(t)$ can be exactly recovered from SFs $\mathbf{s}(t)$ by multiplying both sides of (9) with $\mathbf{W}^{-\mathrm{T}}$

$$\mathbf{x}(t) = \mathbf{W}^{-\mathrm{T}}\mathbf{s}(t) \triangleq \mathbf{R}^{\mathrm{T}}\mathbf{s}(t) \tag{40}$$

where $\mathbf{R} = \mathbf{W}^{-1}$. Here both inputs and SFs are of equal dimensions $\mathbf{u}(t), \mathbf{s}(t) \in \mathbb{R}^m$. As with PSFA, the relationship between q and m can be arbitrary, and the mapping from $\mathbf{s}(t)$ to $\mathbf{x}(t)$ is denoted as

$$\mathbf{x}(t) = \mathbf{H}\mathbf{s}(t)+\mathbf{e}_x(t) \tag{41}$$

when $q \le m$, we can separate (40) into following two parts

$$\mathbf{x}(t) = \mathbf{R}_1^{\mathrm{T}}\mathbf{s}_{1:q}(t)+\mathbf{R}_2^{\mathrm{T}}\mathbf{s}_{(q+1):m}(t) \tag{42}$$

where $\mathbf{R}_1$ and $\mathbf{R}_2$ denote the first q rows and the last $m - q$ rows of $\mathbf{R}$, respectively. $\mathbf{s}_{1:q}(t)$ and $\mathbf{s}_{(q+1):m}(t)$ denote the q slowest features and the $m - q$ fastest features, respectively. As elucidated previously, the first term in (42) describes the dominant varying trends whilst the second term can be seen as noise. Consequently, it is natural to take $\mathbf{R}_1^{\mathrm{T}}$ as an initial guess for matrix $\mathbf{H}$. In addition, the noise term can be estimated as

$$\mathbf{e}_x(t) \approx \mathbf{R}_2^{\mathrm{T}}\mathbf{s}_{(q+1):m}(t) = \mathbf{x}(t)-\mathbf{R}_1^{\mathrm{T}}\mathbf{s}_{1:q}(t) \tag{43}$$

and we can empirically calculate the variance of each element in $\mathbf{e}_x(t)$ as the initial guess of the variance of measurement noise $\{\sigma_i^2,\ 1 \le i \le m\}$. Up to now, we have devised the initialization of EM algorithm based on the solution of linear SFA in a cost efficient way. However, it seems to have two potential concerns:

- The inequality $q \le m$ must hold because the generalized eigenvalue problem in (10) can at most accept m SFs. For PSFA with over-complete features, that is, $q > m$, this strategy becomes, in principle, no longer feasible.
- To infer SFs $\mathbf{s}(t)$, PSFA actually computes the conditional distribution

$$P(\mathbf{s}(t)|\mathbf{x}(1),\mathbf{x}(2),\cdots,\mathbf{x}(t)) \sim \mathcal{N}(\mu_t, \mathbf{P}_t) \tag{44}$$

using the forward recursions in Kalman filter,[32] and SFs are estimated as its mean value μ_t. In other words, all attainable historical inputs are implicitly employed for the purpose of exact inference. However, for linear SFA, the conditional distribution is roughly approximated as $\mathbf{s}(t) = \mathbf{W}^{\mathrm{T}}\mathbf{x}(t)$, completely discounting the effect of all lagged inputs $\{\mathbf{x}(t-1),\mathbf{x}(t-2),\cdots\}$. The underlying assumption of linear SFA is hence physically suboptimal.

Then we show that these two impediments can be effectively circumvented by applying DSFA to smart initializations. On one hand, with lagged data included, the largest number of SFs derived in linear SFA can be increased from m to $m(d+1)$. Therefore, $m(d+1) \ge q > m$ becomes feasible provided that a sufficiently large d is chosen. Given q, the value of d can be determined as $d^* = \lceil q/m-1 \rceil$ to allow for overcomplete features, where $\lceil\cdot\rceil$ denotes the ceiling function. On the other hand, as discussed in DSFA, SFs are inferred using historical process data $\{\mathbf{x}(t-1),\cdots,\mathbf{x}(t-d)\}$ in a period of time, being more physically reasonable than the generic linear SFA without lags. On this occasion, the exact recovery equation in (40) can be further decomposed as

$$\begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}(t-1) \\ \vdots \\ \mathbf{x}(t-d) \end{bmatrix} = \mathbf{R}^{\mathrm{T}}\mathbf{s}(t) = \begin{bmatrix} \mathbf{R}_{10}^{\mathrm{T}} & \mathbf{R}_{20}^{\mathrm{T}} \\ \mathbf{R}_{11}^{\mathrm{T}} & \mathbf{R}_{21}^{\mathrm{T}} \\ \vdots & \vdots \\ \mathbf{R}_{1d}^{\mathrm{T}} & \mathbf{R}_{2d}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{s}_{1:q}(t) \\ \mathbf{s}_{(q+1):m(d+1)}(t) \end{bmatrix} \tag{45}$$

where $\mathbf{R}_{10}^{\mathrm{T}},\cdots,\mathbf{R}_{1d}^{\mathrm{T}}$ are all $(m \times q)$-dimensional matrices, and $\mathbf{R}_{10}^{\mathrm{T}},\cdots,\mathbf{R}_{1d}^{\mathrm{T}}$ are all $(m \times (md+m-q))$-dimensional matrices. Taking $\mathbf{x}(t)$ into consideration only gives

$$\mathbf{x}(t) = \mathbf{R}_{10}^{\mathrm{T}}\mathbf{s}_{1:q}(t) + \mathbf{R}_{20}^{\mathrm{T}}\mathbf{s}_{(q+1):m(d+1)}(t) \tag{46}$$

Along the same line of linear SFA without lags in (43), the initial guess of $\mathbf{H}$ and $\{\sigma_i^2,\ 1 \le i \le m\}$ can be readily obtained via $\mathbf{R}_{10}^{\mathrm{T}}$ and $\mathbf{R}_{20}^{\mathrm{T}}\mathbf{s}_{(q+1):m(d+1)}(t)$.

Soft Sensor Modeling based on SFs

Probabilistic SF regression

Suppose that we have already derived the PSFA model, and SFs $\mathbf{s}(t)$ are estimated as the mean of the posterior distribution

$P(\mathbf{s}(t)|\mathbf{x}(1),\cdots,\mathbf{x}(t),\theta) \sim \mathcal{N}(\mu_t, \mathbf{P}_t)$ using forward recursions. A requisite step next is to establish the regression model based on SFs, termed as probabilistic slow feature regression (PSFR). The slowness principle has been proposed to select a portion of SFs as predictors of quality variables in a preliminary study.[14] Here, we propose an alternative criterion based on correlation for feature selection in regression, and both criteria are adopted to compare their performance.

Slowness-Based SFR. Because SFs with high indices represent short-term noise to some extent, slowness-based SFR (SlSFR) utilizes $M(\leq m)$ slowest features $\tilde{\mathbf{s}}_{1:M}(t) = \{s_1(t),\cdots,s_M(t)\}^{\mathrm{T}}$ as inputs to build a simple linear regression model

$$y(t) = \mathbf{b}^{\mathrm{T}}\tilde{\mathbf{s}}_{1:M}(t) + c + \epsilon \tag{47}$$

where $y(t)$ is the target output, $\mathbf{b} \in \mathbb{R}^M$ and c are regression coefficients and the bias term, and ϵ is the output residual. Coefficients $\mathbf{b}$ and the bias c can be derived based on the ordinary least squares (OLS) algorithm. The number of SFs q and the selected ones M can be determined by means of validation data.

Correlation-Based SFR. For a better utilization of SFs, the correlation coefficient between each SF and output is evaluated, leading to a reordered vector $\tilde{\mathbf{s}}(t)$, in which $\tilde{s}_1(t)$ has the highest correlation with the output. The linear regression model remains the same format as (47).

The entire procedure of above two PSFR-based soft sensor modeling methods can hence be summarized as follows:

Step 1: Train the PSFA model with the EM algorithm, and derive estimations of q SFs $\{s_1(t),\cdots,s_q(t)\}$ based on forward recursion.

Step 2: Select M quality-relevant SFs $\tilde{\mathbf{s}}_{1:M}(t)$ based on either the slowness criterion or the correlation criterion.

Step 3: Down-sample the quality-relevant SFs in accordance with the sampling time of slow-rate quality data, and obtain N pairs of samples $\{\tilde{\mathbf{s}}_{1:M}(t), y(t)\}$ $(t = t_1,\cdots,t_N)$, where $t_1,\cdots,t_N$ are the sampling moments of y.

Step 4: Perform OLS algorithm and train the linear regression model using $\{\tilde{\mathbf{s}}_{1:M}(t), y(t)\}$ $(t = t_1,\cdots,t_N)$.

Online implementation

We have shown that in linear SFA, the mapping from $\mathbf{x}(t)$ to $\mathbf{s}(t)$ is an approximation to the mean of the posterior distribution $P(\mathbf{s}(t)|\mathbf{x}(1),\cdots,\mathbf{x}(t))$. The online estimation of $\mathbf{s}(t)$ can hence be obtained as the mean of the posterior distribution $P(\mathbf{s}(t)|\mathbf{x}(1),\cdots,\mathbf{x}(t))$. According to the celebrated Kalman filter equation, the online estimate of $\mathbf{s}(t)$ is sequentially given by

$$\mathbf{s}(t) = \mathbf{F}\mathbf{s}(t-1) + \mathbf{K}[\mathbf{x}(t) - \mathbf{H}\mathbf{F}\mathbf{s}(t-1)] \tag{48}$$

where $\mathbf{K} \in \mathbb{R}^{q\times m}$ is the gain matrix calculated in an offline manner. The following iterations in forward recursions are carried out[32]

$$\begin{gathered}
\mathbf{P}_{t-1} = \mathbf{F}\mathbf{V}_{t-1}\mathbf{F}^{\mathrm{T}} + \Gamma,\ t \geq 1 \\
\mathbf{V}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_{t-1},\ t \geq 2 \\
\mathbf{K}_t = \mathbf{P}_{t-1}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}_{t-1}\mathbf{H}^{\mathrm{T}} + \Sigma)^{-1},\ t \geq 2 \\
\mathbf{V}_1 = \mathbf{I} - \mathbf{K}_1\mathbf{H}, \\
\mathbf{K}_1 = \mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{H}^{\mathrm{T}} + \Sigma\right)^{-1}
\end{gathered} \tag{49}$$

Matrix $\mathbf{K}_t$ will approach $\mathbf{K}$ when we take the limit $t \to \infty$.

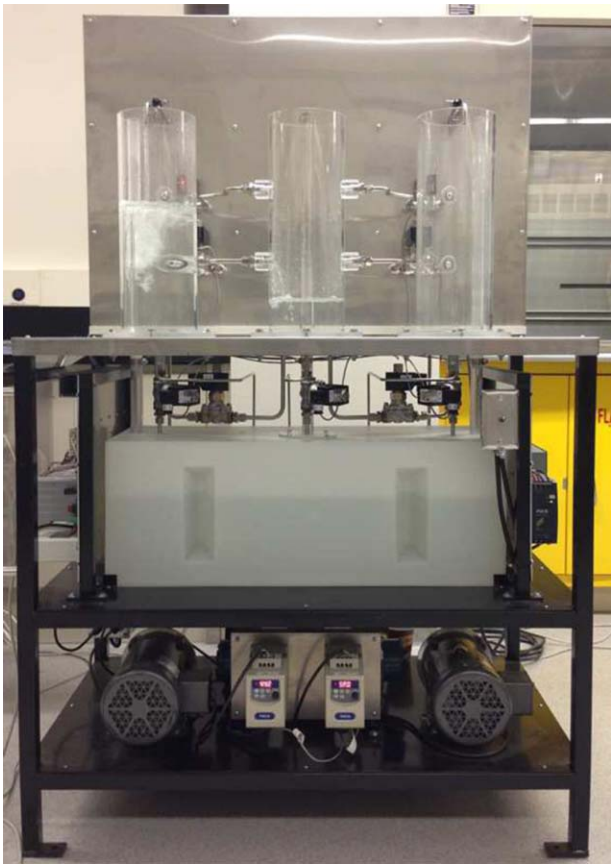

Figure 3. The experimental apparatus of the hybrid tank system.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Implementation Complexity in Practice. For PSFA, only matrices $\mathbf{F}$, $\mathbf{H}$, and $\mathbf{K}$ are required to memorize in online implementation, yielding $(2m+1)q$ parameters in total. Considering regression parameters $\mathbf{b} \in \mathbb{R}^M$ and $c \in \mathbb{R}$, the number of parameters of PSFR to be memorized for online prediction can be calculated as $2mq + q + M + 1$. Therefore, the implementation complexity of PSFR can be denoted as $\mathcal{O}(mq)$. For DPLS and DPCR, its implementation cost is calculated as $\mathcal{O}(md)$ because there are $m(d+1)$ input variables and $m(d+1)$ corresponding coefficients. Therefore, if the process dynamics is extremely slow (e.g., distillation columns) and the product quality becomes heavily influenced by a larger number of lagged variables, the model complexity of DPLS and DPCR will increase rapidly. Roughly speaking, their model complexities grow with the process settling time. In contrast, PSFR is able to accommodate different temporal dynamics by altering the values of transition parameters $\{\lambda_j\}$, resulting in a much more succinct structure with implementation complexity $\mathcal{O}(mq)$. In this sense, PSFR-based models account for process dynamics with a state-space form in a cost efficient way.

Multirate data synthesis

One common limitation for conventional data-driven soft sensors is that fast-rate process samples from distributed control system have to be down-sampled in accordance with quality indices that are sampled slowly, or in some cases, irregularly. In comparison with all fast-rate input samples, the

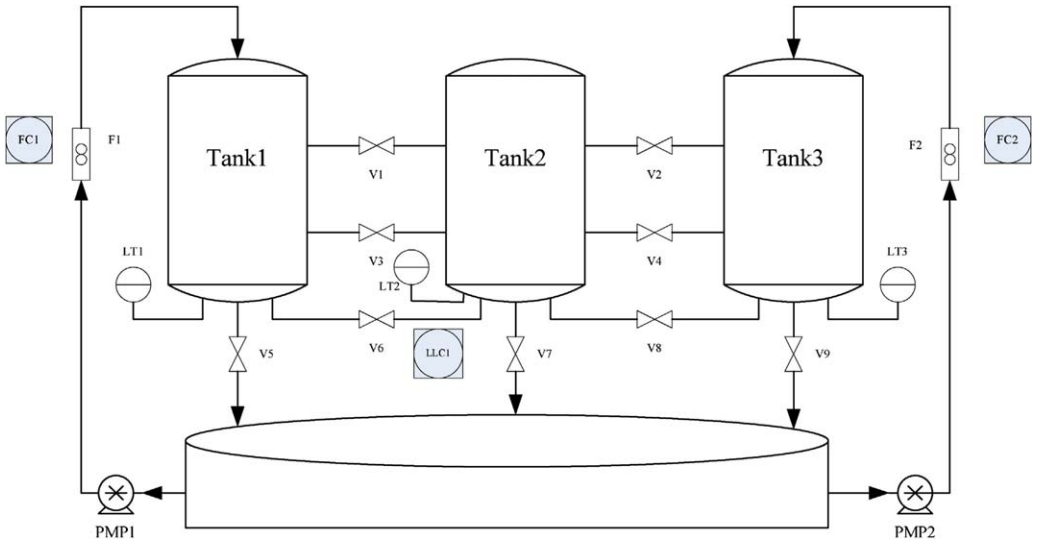

Figure 4. Schematic configuration for the hybrid tank system.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

number of available slow-rate quality samples is much smaller. It hence fails to utilize all meaningful information of available input data.[34] This problem could be well circumvented by the proposed method allowing an unequal number of input and output samples. The success owes to the efficacy of representation learning, that a good unsupervised representation for $P(X)$ is helpful for building the predictive conditional distribution $P(Y|X)$.[8] Several semisupervised soft sensing methods, for instance, PCR[4,35] and deep belief network,[36] are also good examples of representation learning. For PSFR, temporal slowness principle plays an important role because it helps inducing meaningful features from input data, thereby, enabling a desirable multirate data synthesis.

Cases with Nonaligned Sampling Instances of Process Data and Quality Data. In practice, because the quality data are measured via manual laboratory analysis, it is often the case that the sampling instance of quality data cannot be exactly aligned with that of process data. Precisely speaking, if sampling instances of process variables are denoted as integers $\{1,\cdots,T\}$, then the sampling instances of quality variables $\{t_1,\cdots,t_N\}$ are not always ideally integers. It is a challenging task to synthesize process and quality data in this scenario. Synchronizing a quality sample $y(t_k)$ with its closest process sample in the time axis is a pragmatic approximation strategy, which takes effect when the sampling rate of process data is sufficiently fast relative to the process settling time. Otherwise, it will inevitably induce errors to some extent. Fortunately, this pitfall can be overcome using statistical properties of SFs. Next we show how to get the estimation of SFs $\mathbf{s}(t_k)$ at a noninteger sampling moment t_k analytically. Denote t_k^- as the maximal integer that is less than t_k. SFs $\mathbf{s}(t_k^-)$ can be nominally estimated using the posterior distribution $P(\mathbf{s}(t_k^-)|\mathbf{x}(1),\cdots,\mathbf{x}(t_k^-),\theta^{\text{old}}) \sim \mathcal{N}(\mu_{t_k^-},\mathbf{P}_{t_k^-})$, as stated previously. Because SFs evolve as $\mathbf{s}(t)=\mathbf{F}\mathbf{s}(t-1)+\mathbf{e}(t)$, given observations $\{\mathbf{x}(1),\cdots,\mathbf{x}(t_k^-)\}$ only, the SFs at time t_k^-+1 can be estimated as

$$\begin{aligned}
&\mathbb{E}\{\mathbf{s}(t_k^-+1)|\mathbf{x}(1),\cdots,\mathbf{x}(t_k^-)\}\\
=\;&\mathbb{E}\{\mathbf{F}\mathbf{s}(t_k^-)+\mathbf{e}(t_k^-+1)|\mathbf{x}(1),\cdots,\mathbf{x}(t_k^-)\}\\
=\;&\mathbb{E}\{\mathbf{F}\mathbf{s}(t_k^-)|\mathbf{x}(1),\cdots,\mathbf{x}(t_k^-)\}+\mathbb{E}\{\mathbf{e}(t_k^-+1)|\mathbf{x}(1),\cdots,\mathbf{x}(t_k^-)\}\\
=\;&\mathbf{F}\mathbb{E}\{\mathbf{s}(t_k^-)|\mathbf{x}(1),\cdots,\mathbf{x}(t_k^-)\}\\
=\;&\mathbf{F}\mu_{t_k^-}.
\end{aligned} \tag{50}$$

The third equality holds because of $\mathbf{e}(t)\sim\mathcal{N}(\mathbf{0},\Gamma)$ and the independence between $\mathbf{e}(t)$ and observations $\{\mathbf{x}(1),\cdots,$

Table 1. Input Variables for Hybrid Tank

Input No.	Descriptions
1	Liquid level of Tank 1
2	Liquid level of Tank 3
3	Motor speed of Pump 1
4	Motor speed of Pump 2
5	Inlet flow rate of Tank 1
6	Inlet flow rate of Tank 3

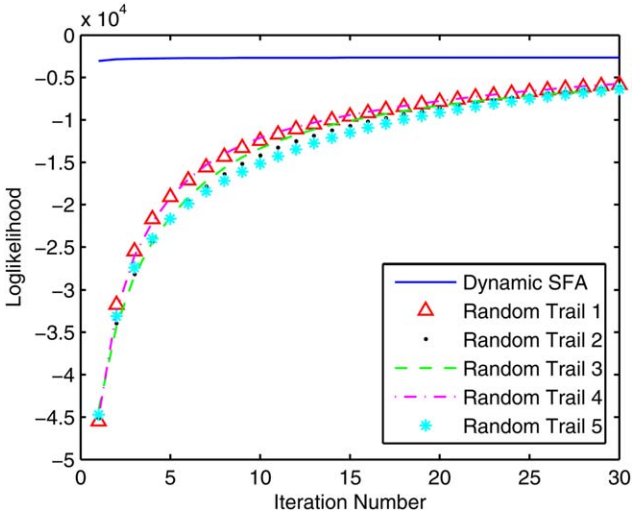

Figure 5. Optimization paths of log-likelihood values in EM algorithm with different initialization strategies ($q=18$).

For random initializations, results of five trials are presented. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

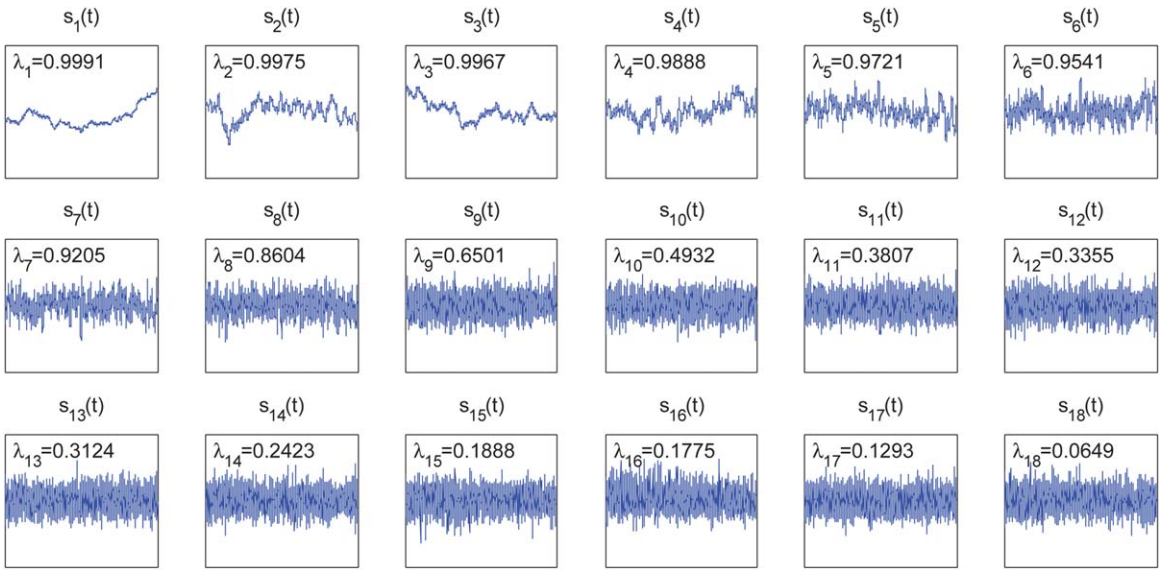

Figure 6. SFs on the training set in the hybrid tank case with $q = 18$.

Time scales of all subfigures are identical. For each SF, its transition parameter λ_j is also reported in each subfigure. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

$\mathbf{x}(t_k^-)\}$. By the same token, we can expect that the estimation of $\mathbf{s}(t_k^- + 2)$ given $\{\mathbf{x}(1), \cdots, \mathbf{x}(t_k^-)\}$ is obtained as $\mathbf{F}^2\mu_{t_k^-}$, and so forth, which decays exponentially with time at integer instances $\{t_k^- + 1, t_k^- + 2, \cdots\}$. It is then natural to generalize this formula to the noninteger instance t_k as a smooth interpolation. Notice that matrix $\mathbf{F} = \text{diag}\{\lambda_1, \cdots, \lambda_q\}$ is diagonal. Therefore, the exponential decay occurs element-wise for each SF, and finally estimations of $\mathbf{s}(t_k)$ can be compactly written as:

$$\mathbb{E}\{\mathbf{s}(t_k)|\mathbf{x}(1), \cdots, \mathbf{x}(t_k^-)\} = \mathbf{F}^{t_k - t_k^-}\mu_{t_k^-}, \tag{51}$$

Table 2. Prediction Results for the Liquid Level of Tank 2 in Hybrid Tank Case

		RMSE	R	Hyper-Parameters
Case A:	DPLS	0.0176	0.8733	$d = 2, A = 12$
$\Delta t = 10$s	SSPDPCR	0.0289	0.6437	$d = 6, A = 15$
	SlSFR	0.0103	0.9575	$q = 18, M = 17$
	CoSFR	**0.0103**	**0.9580**	$q = 18, M = 16$
Case B:	DPLS	0.0166	0.8829	$d = 2, A = 12$
$\Delta t = 20$s	SSPDPCR	0.0291	0.6210	$d = 6, A = 15$
	SlSFR	**0.0089**	**0.9669**	$q = 15, M = 12$
	CoSFR	0.0114	0.9481	$q = 18, M = 15$
Case C:	DPLS	0.0205	0.8387	$d = 2, A = 12$
$\Delta t = 30$s	SSPDPCR	0.0277	0.6231	$d = 6, A = 7$
	SlSFR	0.0109	0.9470	$q = 18, M = 15$
	CoSFR	**0.0105**	**0.9508**	$q = 18, M = 10$
Case D:	DPLS	0.0192	0.8463	$d = 2, A = 12$
$\Delta t = 40$s	SSPDPCR	0.0295	0.6000	$d = 6, A = 13$
	SlSFR	**0.0114**	**0.9425**	$q = 18, M = 18$
	CoSFR	0.0133	0.9243	$q = 18, M = 16$
Case E:	DPLS	0.0184	0.8531	$d = 2, A = 12$
$\Delta t = 50$s	SSPDPCR	0.0249	0.7032	$d = 6, A = 13$
	SlSFR	**0.0080**	**0.9744**	$q = 18, M = 16$
	CoSFR	0.0084	0.9729	$q = 17, M = 17$
Case F:	DPLS	0.0211	0.8420	$d = 2, A = 12$
$\Delta t = 60$s	SSPDPCR	0.0283	0.6742	$d = 5, A = 7$
	SlSFR	**0.0113**	**0.9489**	$q = 15, M = 12$
	CoSFR	0.0126	0.9313	$q = 15, M = 12$

[a]Results in bold are the best in each case.

where $\mathbf{F}^{t_k - t_k^-} = \text{diag}\{\lambda_1^{t_k - t_k^-}, \cdots, \lambda_q^{t_k - t_k^-}\}$.* Then the final linear regression model can be established.

Experimental Case Study: Hybrid Tank System

Experimental settings

In this section, we conduct an experimental study of the PSFR-based soft sensing method using a hybrid tank system in the process control laboratory at the University of Alberta, which is shown in Figure 3. This system has been adopted in a series of research articles for soft sensor modeling as well as change point detection.[37–39] A schematic configuration of the hybrid tank system is provided in Figure 4, which contains three tanks connected in series. Two inlet pumps driven by motors are used to fill water into Tanks 1 and 3. Tank 2 is connected to Tanks 1 and 3 through six valves, namely, V1, V2, V3, V4, V6, and V8. Each tank has its individual outlet valve, namely, V3, V5 and V7. In this study, the valves V1 and V2 are closed and the rest are kept open. Such a configuration leads liquid water in Tanks 1 and 3 to flow into Tank 2 all the time.

In routine conditions, the liquid level of Tank 2 is simultaneously controlled as a primary variable by two cascade controllers, which manipulate the set points of two secondary flow controllers for Tanks 1 and 3. The level of Tank 2 is maintained approximately at 50%. To introduce nominal fluctuations in the operating condition, unmeasured colored noise signals are added to the manipulated variables of speed of two pump motors. In this study, the quality index to be predicted is specified as the level of the middle tank, while six process variables as inputs are listed in Table 1.

For the sake of the multirate phenomenon in practical scenarios, the sampling interval for six process variables is set as 1s, whereas the quality index is sampled based on a longer interval. The system ran regularly for two hours, yielding 7200 fast-rate process samples in total. Data

*For general LDS, it is quite difficult to define $\mathbf{F}^{t_k - t_k^-}$ for an arbitrary matrix $\mathbf{F}$. For PSFA, however, such generalization is unique because $\mathbf{F}$ is diagonal, which particularly applies to this unideal case.

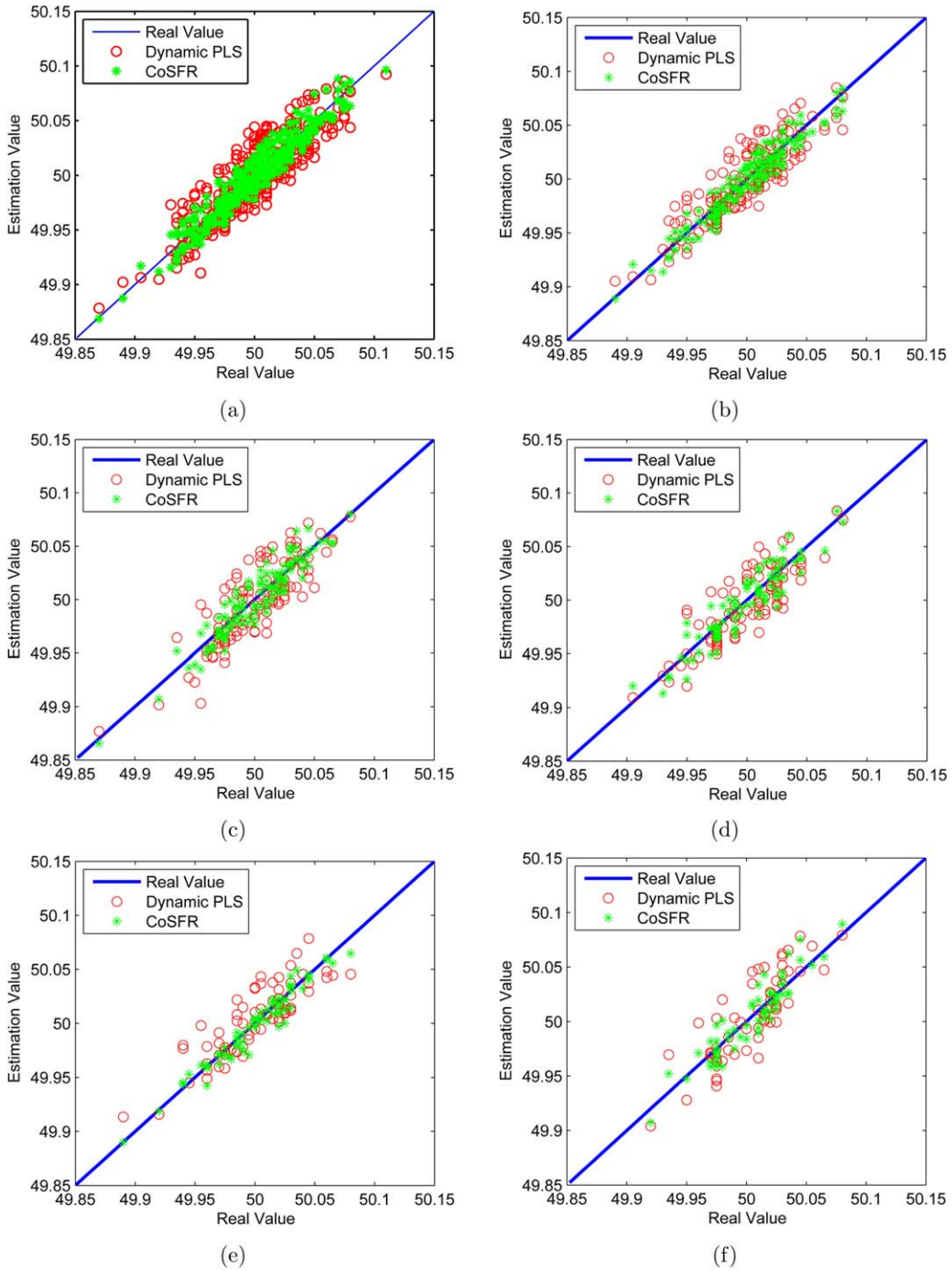

Figure 7. 45 degree comparisons of DPLS and CoSFR in the three tanks case study.

(a) Case A, (b) Case B, (c) Case C, (d) Case D, (e) Case E, (f) Case F. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

samples in the first hour are used for training, and the rest are for test.

Unsupervised SF learning

For the construction of PSFR-based soft sensors, 3600 fast-rate process samples in the training dataset are first used to derive PSFA models. To testify the efficacy of the initialization strategy for PSFA, we use random initializations for EM algorithm for comparison purposes. The results are shown in Figure 5, in which five trials of random initialization are made. It can be obviously seen that the proposed initialization strategy is much more powerful than generic random initializations. On one hand, the initial value obtained by DSFA is very close to the optimized one so that the EM algorithm makes minor modifications. On the other hand, although EM algorithm increases likelihood values of random initializations effectively, their final optimized values could be even poorer than the initial value obtained by DSFA. Because the performance of EM algorithm heavily depends on the initial solution, random initialization is particularly prone to local optima in this scenario, which adds significant difficulty in training PSFA models. The validity of the proposed initialization strategy has thus been demonstrated to furnish a desirable initial solution for parameter optimization and makes EM algorithm

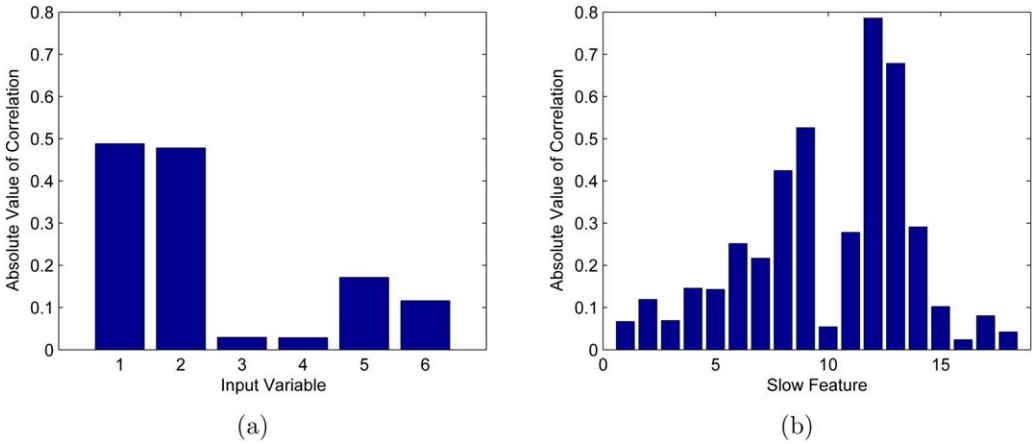

Figure 8. Correlation coefficients with respect to the output in the hybrid tank case.

(a) Absolute values of correlation coefficients between original input variables $\mathbf{x}(t)$ and y. (b) Absolute values of correlation coefficients between SFs $\mathbf{s}(t)$ and y. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

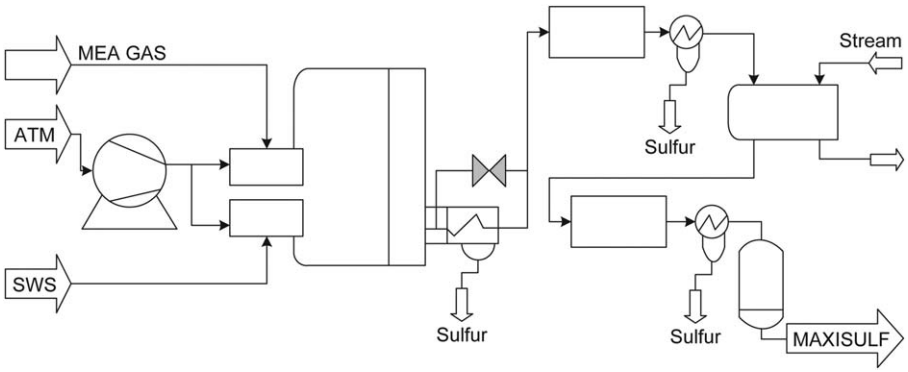

Figure 9. A systematic diagram for SRU process.[41]

computationally thrifty. After establishing the PSFA model using the EM algorithm, estimations of SFs with different slowness are attained. To provide a conceptual picture of SFs, Figure 6 visualizes 3600 SF samples of training data. For each SF, its transition parameter λ_j is also given. It can be clearly seen that their slowness decreases with the index increasing.

Prediction Results and Discussions

For prediction purposes, quality-relevant SFs are selected using two criteria, namely the slowness-based criterion and the correlation-based criterion. Finally, two different linear predictions models, that is, SlSFR and correlation-based SFR (CoSFR), are built with the selected SFs taken as inputs. The hyper-parameters including the number of SFs q and that of quality-relevant ones M, are determined using a portion of latest training data samples as validation data. For performance comparison reasons, DPLS and semi-supervised DPCR (abbreviated as SSPDPCR for simplicity)[35] are adopted in this study because they are the most-used linear regression models that provide allowance for both reduced dimensional subspaces and dynamic issues. For DPLS modeling, same number of process samples and quality samples are used since it is completely supervised. As for SSPDPCR, all available input samples are taken into account, in a similar fashion to two PSFR-based models. Hyper-parameters of DPLS and SSPDPCR like the number of principal components A as well as the number of lags d are selected through five-fold cross-validation on the training data. Six different cases, namely Cases A, B, C, D, E, and F, are considered in which sampling intervals for the quality index are set as 10s, 20s,. . .60s, respectively. The longer the sampling interval, the less available quality samples. For example, in Case A with $\Delta t = 10$s, there are 720 quality samples in the two-hour operation, in which the first 360 samples are used for training and the rest 360 are for test. The prediction results of six cases are reported in Table 2, in which the best ones in terms of accuracy are marked in bold. The root mean square error (RMSE) and the correlation coefficient R are used as evaluation indices of model performances.

As shown in Table 2, DPLS outperforms SSPDPCR in terms of prediction accuracies. Moreover, both PSFR-based models achieve significantly better predictions than DPLS, reducing more than 40% of prediction errors in all cases. Figure 7 further plots the 45 degree comparisons between

Table 3. Input Variables for SRU Process[41]

Input No.	Descriptions
1	MEA gas flow
2	First air flow
3	Second air flow
4	Gas flow in SWS zone
5	Air flow in SWS zone

Table 4. Quality Prediction Results of y_1 in SRU Process

	RMSE	MAE	R	Hyper-Parameters
DPLS	0.0378	0.0216	0.5133	$d = 14, A = 7$
SSPDPCR	0.0381	0.0225	0.5061	$d = 12, A = 8$
SlSFR	0.0351	0.0252	0.5697	$q = 10, M = 8$
CoSFR	0.0300	0.0170	0.6561	$q = 8, M = 4$

Table 5. Quality Prediction Results of y_2 in SRU Process

	RMSE	MAE	R	Hyper-Parameters
DPLS	0.0359	0.0275	0.7431	$d = 4, A = 13$
SSPDPCR	0.0373	0.0287	0.7462	$d = 6, A = 10$
SlSFR	0.0284	0.0224	0.8480	$q = 10, M = 9$
CoSFR	0.0258	0.0206	0.8721	$q = 10, M = 3$

DPLS and CoSFR on six cases. It can be clearly observed that predictions of CoSFR lie much more densely around real values in comparison with those of DPLS. We demonstrate that the inferiority of DPLS is due to its limitation in describing process dynamics. Notice that the hybrid tank system embodies evident dynamics; however, in all cases only two lags are used by DPLS, which is determined justifiably by cross-validation. It has already been reported that DPLS bears the problem of overfitting with an increasing number of lagged variables that are involved, thereby being inadequate in extracting useful information from time series data.[34,40] In contrast, the proposed method is capable to model process dynamics properly by unsupervised learning of fast-rate process data. PSFA suffices to delineate how hidden states $\mathbf{s}(t)$ evolve over time based on a state-space representation, which DPLS and DPCR fail to achieve. In this sense, the dynamic structure of PSFA is physically much clearer than those of DPLS and DPCR.

It is worth mentioning that DPLS and DPCR develop LV subspace in which both input and output information is embodied, whereas LVs of PSFR-based methods are learnt in a purely unsupervised manner. However, even without any output information incorporated, some SFs can provide more useful information for predicting quality index, indicating the slowness principle as a powerful heuristic for inducing quality-relevant representations. Therefore, we shed further light on the merits of SFs by analyzing the relationship between SFs and the model output. As is well-known, if model inputs are highly correlated to the output, a desirable prediction accuracy will be anticipated. Hence data in Case C ($\Delta t = 30$s and $q = 18$) were used to calculate the correlation coefficients between original inputs and output, as well as those between SFs and output, which are depicted in Figure 8. We can see that the first two input variables (liquid levels in Tanks 1 and 3) have some evident correlations (larger than 0.4) with the quality index. However, some of the transformed SFs have much higher correlations, for example, $s_9(t)$, $s_{12}(t)$ and $s_{13}(t)$. It gives evidential insight into the power of slowness principle that, slowness is valid prior knowledge to disentangle underlying driven factors from input data, some of which are beneficial for establishing final predictors. This is the case in which representation learning takes effect. Even though SFs are derived regardless of the output, they benefit supervised modeling because substantial information about the output has been distinctly represented.

Industrial Case Study: SRU Process

In this section we adopt sulfur recovery unit (SRU) process data provided by Fortuna et al.[41] to further illustrate the efficacy of the proposed method. This industrial dataset can be downloaded online,[†] and has been used as a soft sensing benchmark in a number of works.[35,42,43] Two types of gases are taken as inputs of the SRU. The first is MEA gas, which is rich in H_2S; the other one is SWS gas, which is rich in both H_2S and NH_3. The SRU transforms pollutants into pure sulfur through oxidation reaction in reactors. A variety of sensors have been equipped on the plant for online monitoring purposes, in which five variables are selected as relevant inputs to the quality indices, that is, the concentration of H_2S and SO_2 in the tail gas. The process necessitates soft sensors for real-time estimation of concentration of both H_2S and SO_2 to ensure nominal monitoring and control. Therefore, two quality variables $\{y_1, y_2\}$ are to be predicted by soft sensors, and in this study, we build individual prediction model for each quality index. Figure 9 gives a systematic diagram for the SRU process, and the input variables are listed in Table 3.

There are 10,080 process samples sequentially collected based on a sampling interval of 1 min, and each quality index, y_1 or y_2, is sampled based on a longer interval of 30 min, summing up to 336 quality samples in total. The entire dataset is then partitioned into a training set and a test set, each of which contains 5040 fast-rate process samples and 168 slow-rate quality samples.

The prediction results are listed in Tables 4 and 5, in which RMSE, mean absolute error (MAE), and the correlation coefficient R are used as evaluation indices of model performances. It is apparent that in comparison with DPLS and SSPDPCR, both PSFR-based methods generally have improved prediction performances in terms of three indices. Between two PSFR-based methods, CoSFR performs better, and hence we further plot 45 degree comparisons of the predictions of DPLS and CoSFR with reference to the real values in Figure 10. It can be clearly seen that predictions given by CoSFR lie much closer to the diagonal line than those of DPLS.

The absolute values of correlation coefficients between original inputs and outputs $\{y_1, y_2\}$, along with those between SFs and outputs, are reported in Figures 11 and 12. On one hand, the original inputs have low correlations with outputs in both cases, with coefficients less than 0.2. By contrast, PSFA manages to exploit some useful features that are highly correlated to quality variables without resort to output information. For y_1, the 6th, 7th and 8th SFs achieve correlation coefficients about 0.5, and for y_2, the 3rd, 4th and 5th SFs achieve nearly 0.6. On the other hand, some slowest features are irrelevant with quality indices. By targeting on the SFs with high correlations, CoSFR is able to utilize quality-relevant information of SFs more efficiently. It well explains why CoSFR outperforms SlSFR with much fewer regression inputs in this case, as shown in Tables 4 and 5. Note that in the hybrid tank case, SlSFR and CoSFR achieve approximately the same number of quality-relevant features, as well as similar prediction accuracies. This is because the structure of the hybrid tank is relatively simple such that all disturbances are spatially adjacent to the quality index. It is hence reasonable that a large number of SFs be selected in the previous experimental case. For real industrial processes, however, the internal mechanism becomes much more complicated and thus the quality index is likely to be influenced by a fraction of disturbances only. In practical scenarios, CoSFR would be potentially useful to recognize the most important SFs that cause dominant variations of the product quality.

[†]Available at http://www.springer.com/gp/book/9781846284793.

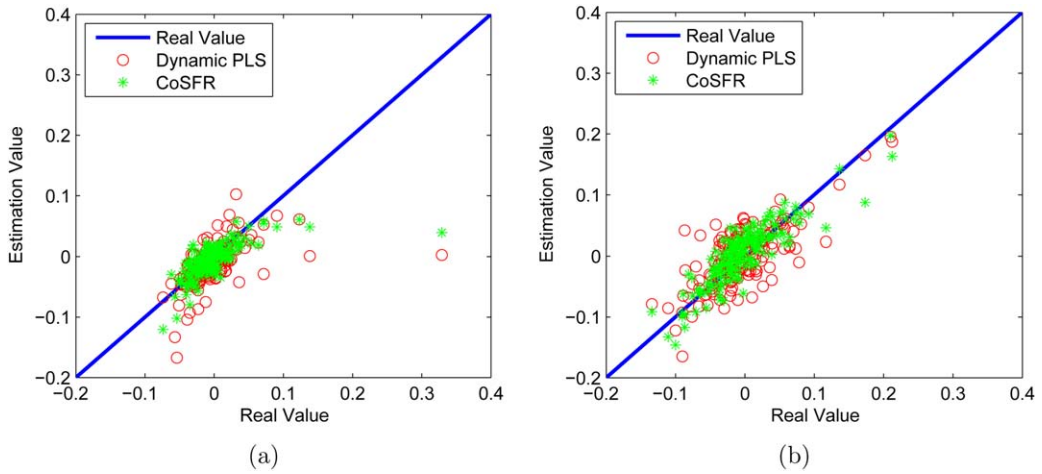

Figure 10. 45 degree comparisons of DPLS and CoSFR.

(a) Results with respect to the first output y_1. (b) Results with respect to the second output y_2. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

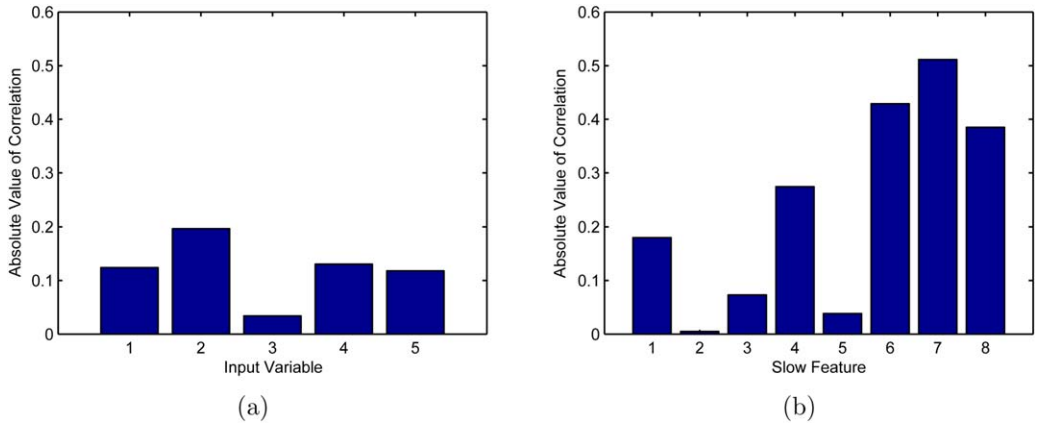

Figure 11. Correlation coefficients with respect to y_1 in the SRU process.

(a) Absolute values of correlation coefficients between original input variables $\mathbf{x}(t)$ and y_1. (b) Absolute values of correlation coefficients between SFs $\mathbf{s}(t)$ and y_1. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Concluding Remarks

In this article, we approach the soft sensing problem from the viewpoint of representation learning. PSFA is adopted to induce meaningful representation of fast-rate inputs in an unsupervised manner, and linear soft sensor models are established using the down-sampled features as inputs. The EM algorithm is adopted to optimize the PSFA model, and a useful strategy is further proposed to improve the EM algorithm based on DSFA. Final predictions are made based on some quality-relevant SFs that are selected. The merits of the

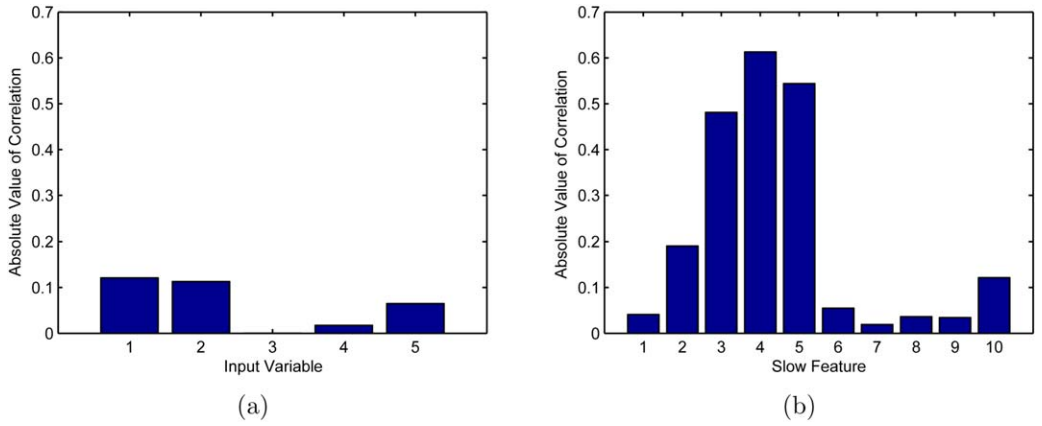

Figure 12. Correlation coefficients with respect to y_2 in the SRU process.

(a) Absolute values of correlation coefficients between original input variables $\mathbf{x}(t)$ and y_2. (b) Absolute values of correlation coefficients between SFs $\mathbf{s}(t)$ and y_2. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

proposed method lie in the following three aspects: (1) partial derived SFs contain meaningful information about the product quality and thus the prediction accuracy can be significantly improved; (2) the development of PSFR models is computationally efficient in face of massive process data, and both fast-rate process data and slow-rate quality data can be reasonably synthesized; (3) process dynamics is compactly represented in a state-space form, which is much simpler than generic dynamic models. This work provides a general representation learning framework for soft sensor modeling problems, and there are also some potential problems for further study, including treatment of nonlinearity and multi-mode characteristics of complex industrial processes.

Acknowledgments

This work was supported by the National Basic Research Program of China (2012CB720505), the National Science Engineering Research Council of Canada (NSERC), Alberta Innovates Technology Futures (AITF), and the National Natural Science Foundation of China (21276137). The first author is grateful for the financial support from China Scholarship Council (CSC). He would also like to thank Zheyuan Liu at the University of Alberta for his help with the experiments.

Literature Cited

1. Kadlec P, Gabrys B, Strandt S. Data-driven soft sensors in the process industry. *Comput Chem Eng*. 2009;33:795–814.
2. Khatibisepehr S, Huang B, Khare S. Design of inferential sensors in the process industry: a review of Bayesian methods. *J Process Control*. 2013;23:1575–1596.
3. MacGregor JF, Cinar A. Monitoring, fault diagnosis, fault-tolerant control and optimization: data driven methods. *Comput Chem Eng*. 2012;47:111–120.
4. Ge Z, Huang B, Song Z. Mixture semisupervised principal component regression model and soft sensor application. *AIChE J*. 2014; 60:533–545.
5. Ge Z, Huang B, Song Z. Nonlinear semisupervised principal component regression for soft sensor modeling and its mixture form. *J Chemometr*. 2014;28:793–804.
6. Dayal BS, MacGregor JF. Recursive exponentially weighted PLS and its applications to adaptive control and prediction. *J Process Control*. 1997;7:169–179.
7. Zamprogna E, Barolo M, Seborg DE. Estimating product composition profiles in batch distillation via partial least squares regression. *Control Eng Practice*. 2004;12:917–929.
8. Bengio Y, Courville A, Vincent PP. Representation learning: a review and new perspectives. *IEEE Trans Pattern Anal Mach Intell*. 2013;35:1798–1828.
9. Qin SJ. Process data analytics in the era of big data. *AIChE J*. 2014; 60:3092–3100.
10. Ku W, Storer RH, Georgakis C. Disturbance detection and isolation by dynamic principal component analysis. *Chemometr Intell Lab Syst*. 1995;30:179–196.
11. Kano M, Miyazaki K, Hasebe S, Hashimoto I. Inferential control system of distillation compositions using dynamic partial least squares regression. *J Process Control*. 2000;10:157–166.
12. Wiskott L, Sejnowski T. Slow feature analysis: unsupervised learning of invariances. *Neural Comput*. 2002;14:715–770.
13. Shang C, Yang F, Gao X, Huang X, Suykens JAK, Huang D. Concurrent monitoring of operating condition deviations and process dynamics anomalies with slow feature analysis. *AIChE J*. 2015; doi: 10.1002/aic.14888.
14. Shang C, Yang F, Gao X, Huang D. Extracting latent dynamics from process data for quality prediction and performance assessment via slow feature regression. In American Control Conference, Chicago, IL, 2015; in press.
15. Körding KP, Kayser C, Einhäuser W, König P. How are complex cell properties adapted to the statistics of natural stimuli? *J Neurophysiol*. 2004;91:206–212.
16. Berkes P, Wiskott L. Slow feature analysis yields a rich repertoire of complex cell properties. *J Vision*. 2005;5:579–602.
17. Franzius M, Wilbert N, Wiskott L. Invariant object recognition with slow feature analysis. In: *Artificial Neural Networks-ICANN 2008*. Berlin, Heidelberg: Springer, 2008, 961–970.
18. Wu C, Du B, Zhang L. Slow feature analysis for change detection in multispectral imagery. *IEEE Trans Geosci Remote Sensing*. 2014;52: 2858–2874.
19. Wu C, Zhang L, Du B. Hyperspectral anomaly change detection with slow feature analysis. *Neurocomputing*. 2015;151:175–187.
20. Blaschke T, Zito T, Wiskott L. Independent slow feature analysis and nonlinear blind source separation. *Neural Comput*. 2007;19:994–1021.
21. Minh HQ, Wiskott L. Multivariate slow feature analysis and decorrelation filtering for blind source separation. *IEEE Trans Image Processing*. 2013;22:2737–2750.
22. Sprekeler H, Zito T, Wiskott L. An extension of slow feature analysis for nonlinear blind source separation. *J Mach Learning Res*. 2014;15:921–947.
23. Tipping ME, Bishop CM. Probabilistic principal component analysis. *J R Stat Soc: Ser B*. 1999;61:611–622.
24. Bechman CF, Smith SM. Probabilistic independent component analysis for functional magnetic resonance imaging. *IEEE Trans Med Imaging*. 2004;23:137–152.
25. Koller D, Friedman N. *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, Massachusetts: MIT Press, 2009.
26. Turner R, Sahani M. A maximum-likelihood interpretation for slow feature analysis. *Neural Computat*. 2007;19:1022–1038.
27. Digalakis V, Rohlicek JR, Ostendorf M. ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition. *IEEE Trans Speech Audio Process*. 1993;1:431–442.
28. Ghahramani Z, Hinton GE. Parameter estimation for linear dynamical systems. Technical Report CRG-TR-96-2, University of Toronto, Dept. of Computer Science, Toronto, Ontario, 1996.
29. Lee TW, Lewicki MS, Girolami M, Sejnowski TJ. Blind source separation of more sources than mixtures using overcomplete representations. *IEEE Signal Processing Lett*. 1999;6:87–90.
30. McLachlan G, Krishnan T. *The EM Algorithm and Extensions*. Hoboken, New Jersey: Wiley, 2007.
31. Jin X, Huang B. Robust identification of piecewise/switching autoregressive exogenous process. *AIChE J*. 2010;56:1829–1844.
32. Bishop CM. *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
33. Kalman RE. A new approach to linear filtering and prediction problems. *J Fluids Eng*. 1960;82:35–45.
34. Shang C, Huang X, Suykens JAK, Huang D. Enhancing dynamic soft sensors based on DPLS: a temporal smoothness regularization approach. *J Process Control*. 2015;28:17–26.
35. Ge Z, Song Z. Semisupervised Bayesian method for soft sensor modeling with unlabeled data samples. *AIChE J*. 2011;57:2109–2119.
36. Shang C, Yang F, Huang D, Lyu W. Data-driven soft sensor development based on deep learning technique. *J Process Control*. 2014; 24:223–233.
37. Chen L, Tulsyan A, Huang B, Liu F. Multiple model approach to nonlinear system identification with an uncertain scheduling variable using EM algorithm. *J Process Control*. 2013;23:1480–1496.
38. Keshavarz M, Huang B. Bayesian and expectation maximization methods for multivariate change point detection. *Comput Chem Eng*. 2014;60:339–353.
39. Keshavarz M, Huang B. Expectation Maximization method for multivariate change point detection in presence of unknown and changing covariance. *Comput Chem Eng*. 2014;69:128–146.
40. Shang C, Yang F, Gao X, Huang D. A comparative study on improved DPLS soft sensor models applied to a crude distillation unit. In: *Proceedings of International Symposium on Advanced Control of Chemical Processes*, Whistler, Canada, 2015; in press.
41. Fortuna L, Graziani S, Rizzo A, Xibilia MG. *Soft sensors for monitoring and control of industrial processes*. London: Springer, 2007.
42. Kadlec P, Grbić R, Gabrys B. Review of adaptation mechanisms for data-driven soft sensors. *Comput Chem Eng*. 2011;35:1–24.
43. Xie L, Zeng J, Gao C. Novel just-in-time learning-based soft sensor utilizing non-Gaussian information. *IEEE Trans Control Sys Technol*. 2014;22:360–368.

Manuscript received Apr. 9, 2015, and revision received May 23, 2015.