IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 November 2023, accepted 17 December 2023, date of publication 19 December 2023,
date of current version 16 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3344932

RESEARCH ARTICLE

A Unifying View of Multivariate State Space Models for Soft Sensors in Industrial Processes

WENYI LIU AND TAKEHISA YAIRI, (Member, IEEE)
Department of Advanced Interdisciplinary Studies, The University of Tokyo, Tokyo 153-8904, Japan

Corresponding author: Wenyi Liu (liu-wenyi@g.ecc.u-tokyo.ac.jp)

The work of Wenyi Liu was supported by the China Scholarship Council.

ABSTRACT State-space formulations offer a flexible approach for developing soft sensors in industrial processes, leveraging both data information and domain knowledge of process dynamics. On one hand, the state vector introduces varying perspectives in modeling process dynamics. However, choosing the definition of a state vector that is appropriate for the data and problem at hand is not a simple task. In this study, we examine and bridge three hybrid models using the framework of state space equations. We explore three key aspects within this framework: problem formulation, state prediction, and parameter estimation by the Expectation-Maximization (EM) algorithm. We compare the three hybrid models and two recurrent neural networks (RNN) approaches on three real-world datasets from desulfuring, polymerization, and sulfur recovery processes. Results are analyzed from both the data perspective and the process perspective, aiming to enhance the understanding and implementation of soft sensors in dynamic settings, with potential implications for various industries relying on accurate and adaptable soft sensor technologies.

INDEX TERMS Auto-regressive dynamic latent variables (ADLV), linear dynamical system (LDS), quality prediction, multivariate time series, soft sensor, state space models, structural time series (STS).

I. INTRODUCTION

When building soft sensors, a single static model is often inadequate to describe the observed data, as the system is susceptible to various changes such as mechanical element abrasion, shifts in operating modes, process faults, material quality variations, and weather fluctuations, among other factors. Such system changes are directly reflected in the distribution divergence between the training data and test data, posing serious challenges for real-world applications: for example, the degradation of a trained model after a period of online operation. Therefore, dynamic models have shown to be more practical and realistic for updating and maintaining soft sensors in non-stationary environments [1], [2], [3], [4], [5].

The combination of conventional multi-statistical algorithms (e.g., PCR, SVR, PLS, ANN) with adaptive learning has emerged as a popular solution to this problem. These approaches incrementally update models to react to changing environments during online operations. Model adaptation can be achieved in two ways: 1) by incorporating more recent samples through a moving window or forgetting factor [3], and 2) by utilizing more relevant and similar data based on distance or density distribution, such as just-in-time-learning [4] and importance weighting [5].

A common characteristic of these approaches lies in addressing the drift adaptation problem from a data perspective. However, it is essential to recognize that data drift is a consequence rather than the root cause itself [6]. Simply retraining and updating the local models in response to data changes does not address the underlying reasons behind the drift. These approaches lack physical interpretations and may face challenges in gaining the trust of domain experts.

In contrast, the state-space approach, as an alternative solution for maintaining soft sensors, does not suffer from these drawbacks. State-space models (SSM) are inherently model-oriented, allowing the incorporation of process expert knowledge into the model, such as process dynamics and measurement noise. In industrial processes, SSM has proven to be a flexible and robust framework for representing and controlling dynamic systems implemented by First Principle Models (FPM) [7], controller design [8] and

The associate editor coordinating the review of this manuscript and approving it for publication was Ming Xu.

system identification of multiple input multiple output systems [9].

To develop soft sensors with SSM, the definition of the state vector plays a significant role in modelling the process dynamics. Due to the flexibility of the SSM framework, there are many different ways to define state vectors. Most studies in this field have focused only on specific approaches, such as dynamic latent variable models [17], [18], [19], [20], [21]. In this case, the state vector refers to the extracted latent feature space. However, it can also describe the regression coefficients or other variables of interests. Therefore, it would be interesting to investigate whether these different methods result in similar performance in terms of predictions or their performance depend on the applications and data at hand. So far, here has been little comparative analysis of the model performance in different scenarios and across different definitions of the state vector.

In this article, we discuss and compare three representative solutions from a practical standpoint. By investigating and analyzing these different formulations, we aim to explore their respective strengths, limitations, and potential applications in soft sensor development. We believe that these comparisons offer a broader perspective on modelling process dynamics to both researchers and practitioners. The contributions of this paper are as follows:

1) We bridge and connect the time-varying coefficient models for dynamic soft sensors, the auto-regressive latent variable models for process data analysis, and decomposition and separation of time series in econometrics under the framework of SSM.
2) We provide solutions for the general case to unify various models, in particular, with regard to dynamics modelling, parameter inference and estimation, and prediction. Specifically, Kalman filter and smoother are utilized for state inference and prediction, which help build more interpretable soft sensors.
3) The key parameters required are estimated iteratively by the Expectation Maximization (EM) algorithm, which aides model development when no prior knowledge is available.
4) We present results on three case studies, and the differences in performance of the models are explained and discussed from both the data perspective and process perspective.

II. RELATED WORK

Before proceeding with the literature, it is essential to acknowledge that the state-space formulations for data-driven soft sensors are not unique, and the assumption of the presence of a "true" model is unrealistic. In practice, a "good" model is often more desirable in terms of prediction performance. Constructing a suitable model requires one to consider empirical facts, domain knowledge, historical context, data quality, and research objectives. These insights help identify the dynamic factors of a system, including but not limited to: 1) time-varying parameter models, implying gradual changes of the linear regression coefficients; 2) autoregressive latent variables projected from higher-dimensional inputs that are potentially contaminated by random noise; 3) a separate additive nonstationary disturbance influencing a stationary regression model.

A. LINEAR DYNAMICAL SYSTEMS (LDS)

The first case focuses primarily on extening static soft sensors into dynamic models by allowing time-varying parameters through the Kalman filter. These models are commonly referred to as linear Gaussian state space models [10], whereas more specifically, they are also known as the linear dynamical systems (LDS) [11] or simply time-variant soft sensors [12]. For consistency, we will use the term LDS to represent these types of models. The concept of LDS has been adopted in many applications. Xu et al. employed the LDS to reflect the personalized time-varying treatment effects for Pakinson's disease patients [13]. Dastjerd et al. present a novel algorithm that combines generalized random walk, multi-state-dependent parameter, and time varying parameter to achieve high accuracy online quality monitoring [14]. Liang et al. applied this approach to an electrical/ultrasonic dual-modality dynamic imaging method to reconstruct the time-varying distribution [15]. While the assumption of LDS is attractive and convenient, it can sometimes be too general and vague in meaning, limiting its application. Therefore, it is necessary to complement the soft sensor development with other perspectives.

B. DYNAMIC LATENT VARIABLES (DLV)

In the second scenario, emphasis is shifted towards modeling latent variable with explanatory variables. On one hand, dimensionality reduction is often necessary to deal with the high multicollinearity and redundancy in process data. On the other hand, process data might be corrupted by random noises or other nonstationary disturbances [16]. These two factors collectively lead to the dynamic latent variable methods. For example, Wen et al. extend the traditional PCA to a linear Gaussian state-space model to allow dynamics of the latent scores [18]. Similarly, Li et al. develop a dynamic PLS that incorporating the supervision into the measurement equation [19]. Qin et al. has provided a comprehensive comparison of these data-based latent dynamic variable models for prediction and monitoring [21]. Due to the roots in traditional dimension reduction techniques, it is observed that these types of methods often model the measured inputs and measured outputs separately by the latent scores, resulting in three equations. In this study, we integrate the dimension reduction into the state equation, aiming for more direct comparison with the time-varying coefficient models.

Moreover, we consider Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) [22] as relevant studies to DLV models. RNNs typically define hidden units as a function of both previous units and the current input, indicating a similarity to the operation of

SSM. Medsker et al. [23] present and study the architecture of RNNs with classic state-space representations from a dynamic system perspective. Although in principle RNNs are rooted in modelling dynamic systems, their structures of memory cells and gates involve nonlinear operations and pose challenges from dealing with data shortage, model training, parameter tuning, and vanishing and exploding gradients [24].

C. DECOMPOSITION OF TIME SERIES

The last circumstance draws inspiration from the applications of SSM in economic and financial research, where time series are separated into several explainable and simpler components (trends, seasonalities, trading day effects, human errors, etc) [25]. This splitting-based time series modeling approach has been applied in many domains. R. Salles et. al review and compare several transformation methods that separate nonstationary time series to explain the intrinsic physical phenomena such as deterministic trends and structural breaks [26]. Stathopoulos et al. propose a traffic model that supplements the observed traffic flow with other factors (roadway capacity, section length, signalization plans, weather, etc.) [27]. Liu et al. [28] present DynaConF to decouple stationary conditional distribution from nonstationary part, and it has demonstrated better performance than some state-of-the-art deep learning methods on several public datasets.

Building on this idea, the regression of sensor data, which are commonly used as the exogenous variables by data-driven soft sensors, can be viewed as a global stationary component. And this major effect may influence the target variable with other additive nonstationary components together (such as self-evolving stochastic noise and control variables). This new perspective offers opportunities to address complex changes and dynamics of the industrial systems.

Although individual efforts have been made to explore each formulation, to our best knowledge, there has been little attention paid to their comparisons. We aim to provide a comprehensive and unified view of dynamic soft sensor development within the framework of SSM. We thoroughly discuss, analyze and compare their differences in assumptions, formulation, and parameter estimation among these models. Through this comparative analysis, we seek to gain insights into the dynamic modeling problems in real-world applications.

III. METHODOLOGY

A. COMMON FRAMEWORK

We first present the general framework for soft sensors, and it includes the soft sensor formulation with SSM, the Kalman filter and smoother for inference, and the EM algorithm for parameter learning.

1) SSM FOR SOFT SENSORS

The most general state-space representation of soft sensors with m inputs, n outputs, and d state variables can be written in the following form:

$$\text{(State equation) } x_{k+1} = Ax_k + Bu_k + w_k \tag{1}$$

$$\text{(Observation equation) } y_k = Cx_k + Du_k + v_k \tag{2}$$

$$\begin{pmatrix} w_k \\ v_k \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix}\right), \tag{3}$$

where $k = 1, 2, \cdots, n$ is the sample number, $x_k \in \mathbb{R}^d$ is the state vector, and $u_k \in \mathbb{R}^m$ and $y_k \in \mathbb{R}^n$ are the measured input and output vector at time instant k, respectively. The vector $w_k \in \mathbb{R}^d$ and vector $v_k \in \mathbb{R}^n$ are unrelated vector white noise with mean zero and covariance matrices Q and R. $A \in \mathbb{R}^{d\times d}$ is the state transition matrix modelling the dynamics of the physical system, $B \in \mathbb{R}^{d\times m}$ is the input matrix, $C \in \mathbb{R}^{n\times d}$ is the emission matrix relating the measurements to the state, and the remaining matrix $D \in \mathbb{R}^{n\times m}$ is the direct transition matrix.

This general form of SSM defined by (1) and (2) together can describe a rich class of dynamic models. In the context of soft sensor modelling, this is also a very powerful time series model with widespread applications. Under this problem formulation, y_k can be understood as the quality variable, or the difficult-to-measure variable, such as the flash point of a petroleum or the concentration of a chemical. u_k, the fixed input, can be seen as the exogenous or predetermined variables. Usually, it refers to the easy-to-measure data, or more directly, features, that may enter into the state equation or observation equation. However, the definition of x_k is much more flexible and is therefore one of the key research focus of this study.

According to the above formulation, the transition equation and observation equation define a Markov chain

$$p(y_{1:n}, x_{1:n}) = p(x_1)p(y_1|x_1, u_1) \prod_{k=2}^{n} p(x_k|x_{k-1}, u_{k-1})p(y_k|x_k, u_k), \tag{4}$$

where the initial vector x_1 is assumed to has a mean of μ_0 and a covariance Σ_0, and the transition distribution is given by $p(x_k|x_{k-1}, u_{k-1}) \sim \mathcal{N}(x_k|Ax_{k-1} + Bu_{k-1}, Q)$ and emission distribution by $p(y_k|x_k, u_k) \sim \mathcal{N}(y_k|Cx_k + Du_k, R)$, respectively. In this model, there are some important parameters, denoted by $\theta \overset{def}{=} \{\mu_0, \Sigma_0, A, B, Q, C, D, R\}$, that often cannot be known in advance. Accordingly, the EM algorithm is employed to determine θ as well as the unknown states. It should be noted that, for soft sensor applications, we are particularly interested to predict the next latent state x_k and its corresponding observation y_k given y_1 to y_{k-1}. Therefore, the estimation of EM algorithm serves as a means to deliver better models for prediction and forecasting.

2) EXPECTATION STEP

Let $Y_n = \{y_1, y_2, \ldots, y_n\}$, $X_n = \{x_0, x_1, x_2, \ldots, x_n\}$ and $U_n = \{u_1, u_2, \ldots, u_n\}$, and based on the maximum likelihood estimation, the complete data log-likelihood function can be written as follows:

$$\begin{aligned} l_{Y_n,X_n}(\theta) = &-\frac{1}{2}\ln|\Sigma_0| - \frac{1}{2}(x_0-\mu_0)\Sigma_0^{-1}(x_0-\mu_0)^T \\ &- \frac{n}{2}\ln|Q| - \frac{1}{2}\sum_{k=1}^{n}(x_k - Ax_{k-1} - Bu_{k-1})Q^{-1} \\ &\times (x_k - Ax_{k-1} - Bu_{k-1})^T \\ &- \frac{n}{2}\ln|R| - \frac{1}{2}\sum_{k=1}^{n}(y_k - Cx_k - Du_k)R^{-1} \\ &\times (y_k - Cx_k - Du_k)^T. \end{aligned} \tag{5}$$

In the *E-step*, the objective is to make inference about the local posterior marginals for the latent variables from the observed sequence, and the conditional expectation of $l_{Y_n,X_n}(\theta)$ given all the observed data Y_n and a parameter set θ^{old} is denoted as

$$Q(\theta|Y_n, \theta^{\text{old}}) = \mathbb{E}[l_{Y_n,X_n}(\theta)|Y_n, \theta^{\text{old}}]. \tag{6}$$

The EM algorithm for SSM requires the smoothed estimates of x_k, which can be obtained through Kalman filter and Kalman smoother. For convenience, we denote the conditional expectation of state x_k given the observation up to time s as $x_k^s = \mathbb{E}[x_k|Y_s]$, and the corresponding conditional expectation of the variance of estimation error is $P_k^s = \mathbb{E}[(x_k - x_k^s)(x_k - x_k^s)^T|Y_s]$. Similarly, the covariance matrix of error at time k and t is expressed as $P_{k,t}^s = \mathbb{E}[(x_k - x_k^s)(x_t - x_t^s)^T|Y_s]$.

The equations for estimating x_k^s where $s \le n$ are consistent with the Kalman filter equations. With initial conditions $x_0^0 = \mu_0$ and $P_0^0 = \Sigma_0$, and for k=1,2,…, n,

$$\begin{aligned} x_k^{k-1} &= Ax_{k-1}^{k-1} + Bu_{k-1} &\quad (7)\\ P_k^{k-1} &= AP_{k-1}^{k-1}A^T + Q &\quad (8)\\ x_k^k &= x_k^{k-1} + K_k(y_k - Cx_k^{k-1} - Du_k) &\quad (9)\\ P_k^k &= (I - K_kC)P_k^{k-1} &\quad (10) \end{aligned}$$

with

$$K_k = P_k^{k-1}C^T(CP_k^{k-1}C^T + R)^{-1}. \tag{11}$$

The updated mean x_k^k can be seen as taking the predicted mean x_k^{k-1} and subsequently incorporating a correction proportional to the discrepancy between the predicted value $(Cx_k^{k-1} + Du_k)$ and the actual observation y_k. The factor for this adjustment is represented by the Kalman gain matrix K_k. The relevant scales of Q and R are essential for determining the Kalman gain.

Then the posterior marginal distributions x_k^n can be updated conditioned on all the observations Y_n. In the context of time-series data, this involves incorporating both the past and future observations. Although this approach is not applicable for real-time predictions, it serves a crucial role in determining the model's parameters. With initial conditions x_k^k and P_k^k obtained via (7) to (10), for $k = n, n-1, \ldots, 1$,

$$\begin{aligned} x_{k-1}^n &= x_{k-1}^{k-1} + J_{k-1}(x_k^n - x_k^{k-1}) &\quad (12)\\ P_{k-1}^n &= P_{k-1}^{k-1} + J_{k-1}(P_k^n - P_k^{k-1})J_{k-1}^T &\quad (13) \end{aligned}$$

where

$$J_{k-1} = P_{k-1}^{k-1}A^T(P_k^{k-1})^{-1}. \tag{14}$$

The covariance matrix of estimation error at time k and $k-1$ is given by

$$P_{k,k-1}^n = J_{k-1}P_k^n. \tag{15}$$

3) MAXIMIZATION STEP

The *M-step* constitutes maximizing $Q(\theta|Y_n, \theta^{\text{old}})$ with respect to the parameters in θ, and thus produces the updated hypothesis of the parameters, denoted θ^{new}.

Maximizing with respect to μ_0 and Σ_0 gives

$$\begin{aligned} \mu_0^{\text{new}} &= x_0^n &\quad (16)\\ \Sigma_0^{\text{new}} &= P_0^n &\quad (17) \end{aligned}$$

Optimization of A, B, and Q gives

$$\begin{aligned} \begin{bmatrix} A^{\text{new}} & B^{\text{new}} \end{bmatrix} &= \begin{bmatrix} S_{xb} & S_{xz} \end{bmatrix} \begin{bmatrix} S_{bb} & S_{bz} \\ S_{zb} & S_{zz} \end{bmatrix}^{-1} &\quad (18)\\ Q^{\text{new}} &= \frac{1}{n}\{S_{xx} - S_{xb}(A^{\text{new}})^T - S_{xz}(B^{\text{new}})^T \\ &\quad - A^{\text{new}}S_{bx} + A^{\text{new}}S_{bb}(A^{\text{new}})^T \\ &\quad + A^{\text{new}}S_{bz}(B^{\text{new}})^T \\ &\quad - B^{\text{new}}S_{zx} + B^{\text{new}}S_{zb}(A^{\text{new}})^T \\ &\quad + B^{\text{new}}S_{zz}(B^{\text{new}})^T\} &\quad (19) \end{aligned}$$

and the closed solutions for A^{new} and B^{new} are provided in (45) and (46).

Maximization with respect to C, D, and R leads to

$$\begin{aligned} \begin{bmatrix} C^{\text{new}} & D^{\text{new}} \end{bmatrix} &= \begin{bmatrix} S_{yx} & S_{yu} \end{bmatrix} \begin{bmatrix} S_{xx} & S_{xu} \\ S_{ux} & S_{uu} \end{bmatrix}^{-1} &\quad (20)\\ R^{\text{new}} &= \frac{1}{n}\{S_{yy} - S_{yx}(C^{\text{new}})^T - S_{yu}(D^{\text{new}})^T \\ &\quad - C^{\text{new}}S_{xy} \\ &\quad + C^{\text{new}}S_{xx}(C^{\text{new}})^T + C^{\text{new}}S_{xu}(D^{\text{new}})^T \\ &\quad - D^{\text{new}}S_{uy} \\ &\quad + D^{\text{new}}S_{ux}(C^{\text{new}})^T + D^{\text{new}}S_{uu}(D^{\text{new}})^T\} \\ &&\quad (21) \end{aligned}$$

and the closed solutions for C^{new} and D^{new} are provided in (47) and (48). The above solutions have used the smoothed estimates, covariances, and cross moments. Some of them are replaced by symbols for simplicity, see (34) to (44).

The *E-step* and *M-step* are repeated until convergence. Convergence can be referred to as either the algorithm reaches a predefined number of iterations or the log-likelihoods of two iterations differ by a predefined threshold. We adopt the first condition in this study.

B. MODEL-SPECIFIC DESCRIPTIONS

In the last section, we have introduced the inference and learning in the most general state-space formulation. Next, we discuss three specific modelling approaches that having differing assumptions and formulations.

1) LDS

Unlike traditional fixed-parameter soft sensors, LDS allows the regression coefficients to vary with time. The state-space representation for LDS follows

$$\text{(State equation) } x_{k+1} = Ax_k + w_k \tag{22}$$

$$\text{(Observation equation) } y_k = C_k x_k + v_k \tag{23}$$

where x_{k+1} is the regression parameter at time $k+1$. The evolution of x_{k+1} is controlled by the deterministic and linear transformation of x_k to Ax_k, and followed by a random walk w_k. Then, C_k, the measured sensor data at time k, relates the observation y_k to the state variable x_k. The transition matrix A in this case is often assumed to be a diagonal matrix for simplicity [1], [20]. In some works, A is assumed to be identity matrix, modelling small variation and gradual change of the regression parameter [30].

It can be seen that (23) has the structure of a linear regression model but the coefficient vector x_k varies over time. A simple illustration of LDS is provided in the left part of Fig. 1, where the wavy line depicts the time-varying coefficient vector. Generally, soft sensors with constant coefficients may work for some special static and stationary cases, but models with time-variant regression parameter are more realistic in reality. Not only is LDS a powerful tool to analyze a wide range of system changes occurred in dynamical physical systems, it also provides additional facilities such as incorporating prior knowledge of process dynamics and data information, retaining tractability of inference, and requiring no predefined window size or forgetting factor.

Due to the simple form of the LDS model, its parameter set $\theta \overset{def}{=} \{\mu_0, \Sigma_0, A, Q, R\}$ can be easily obtained by imposing 0 on all elements of B and D.

2) ADLV

Different from the LDS models, the state variables in ADLV models are defined in terms of the feature space, rather than the regression coefficient. In this study, the state-space representation for ADLV models follows

$$\text{(State equation) } x_{k+1} = Ax_k + Bu_k + w_k \tag{24}$$

$$\text{(Observation equation) } y_k = Cx_k + v_k \tag{25}$$

where the state equation describes the dynamic latent variables. The latent variable x_{k+1} not only has a correlation structure originated from the multivariate autoregressive model, where the correlation is captured by the transition matrix A, and it is also related to the linear projection of the measured input u_k. w_k is the white Gaussian noise. The measurement y_k is a linear projection from a lower dimension of current latent feature x_k, where C is the regression parameter and the noise v_k is the normal distributed residual.

This formulation is distinguished with some other dynamic latent variable models proposed in [18], [19], [20], and [21] that it directly incorporates the measured input into the state equation. Furthermore, it is noted that Zhou et al. [29] have used this terminology previously, i.e., the autoregressive dynamic latent variable (ARDLV) model, they argue that the latent variable at current time step is correlated to its past L values, and thus, resulting a AR(L) process. However, cross correlations are not considered in this study and our model remains an AR(1) process.

It can be observed that through the introduction of u_k, the state equation of ADLV model has more parameters than the corresponding LDS model, and it is also less straightforward to understand the model. The middle part of Fig. 1 provides a simple illustration of the ADLV models, where the circles (or ellipses) with concentric rings symbolize the expanding and contracting latent feature space, and the arrows radiating in and out illustrate the potential changes and dynamics of the system. By imposing zeros on D, the solution of parameter estimations of the ADLV models ($\theta \overset{def}{=} \{\mu_0, \Sigma_0, A, B, Q, C, R\}$) through EM algorithm follows the general derivations in Section III-A2 and Section III-A3.

3) STS

Although the majority of the study on this topic define the state vector as either related to the regression coefficients or the latent feature space, it is possible to attribute the process dynamics to other factors. In fact, there are advantages to include all the dynamic variables in the state vector so that the measured input can be seen as deterministic. For example, we define a simple structural time series (STS) model as the following

$$\text{(State equation) } x_{k+1} = Ax_k + w_k \tag{26}$$

$$\text{(Observation equation) } y_k = Cx_k + Du_k + v_k \tag{27}$$

where the process dynamics at time $k+1$ are assumed to be captured by the state vector x_{k+1}, and y_k is a linear transformation of x_k with the addition of a classical linear regression on the known or predetermined exogenous vector u_k, and an observation noise modeled by v_k.

The essence of the STS models lies in the decomposition of the observations into several components, and domain knowledge plays an important role. For example, (26) can represent a known FPM expressed in a difference equation (or a set of equations) [7]. The state vector can also account for external factors that influence the process, such as the change of the weather or season, the occurrence of system maintenance, and human error, which can enter the observation equation in a similar manner as (26). This is parallel in structure with time series modelling in economic research, which seeks explanations of the observation from a variety of effects, such as the stochastic trend, the seasonal component, rare event, etc. Thus, not only does the STS

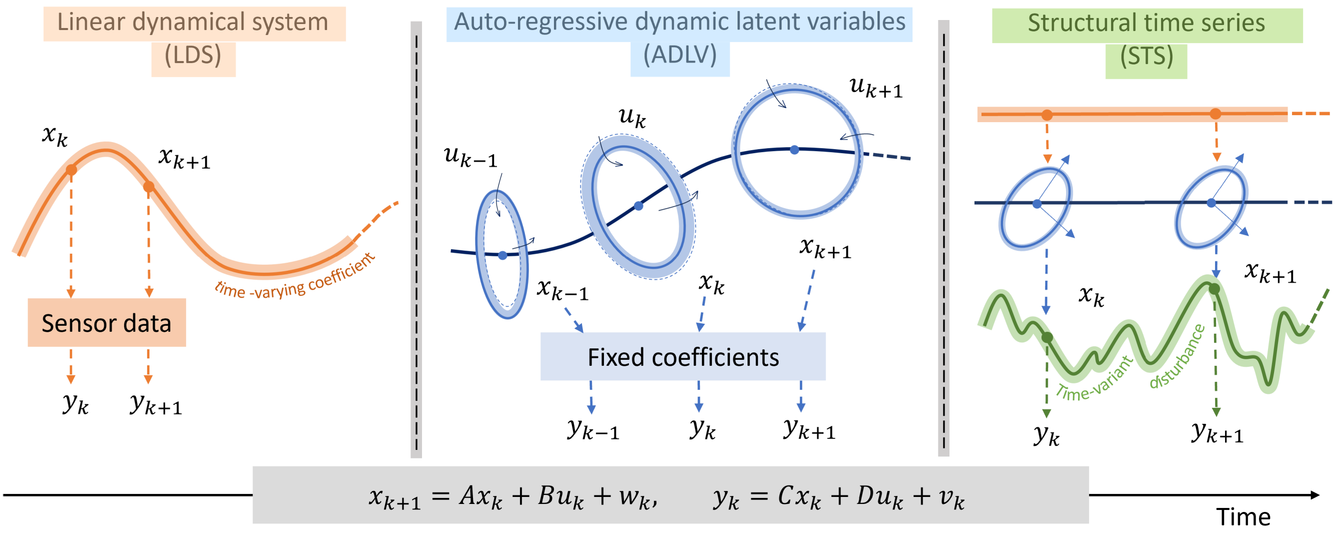


FIGURE 1. Simplified graphic illustrations of the three dynamic models that all function under the standard state-space framework: *left*) the LDS model; *middle*) the ADLV model; *right*) the STS model.

model support the specification of the model, but it also offers more interpretability and assists further analysis.

In Fig. 1, the illustration of STS model (right) is on the basis of the LDS models and the ADLV models, where neither change of the regression coefficients (straight line) nor fluctuations of the feature space are observed. On the contrary, some additional disturbance influences the process dynamics.

To sum up, the three examples we have listed model the process dynamics from different point of views, and details regarding the descriptions of the variables and matrices are summarized in Table 1.

TABLE 1. The model-specific differences in descriptions.

Model	X_n	U_n	B	C	D
DLS	coefficients	0	0	sensor data	0
ADLV	latent features	sensor data	$\mathbb{R}^{d\times m}$	coefficients	0
STS	disturbances	sensor data	0	$\mathbb{R}^{n\times d}$	coefficients

C. PREDICTION AND FORECASTING

After learning the parameters through EM algorithm, the models could be readily used for prediction. In this study, we investigate both the online prediction and offline prediction of the three models. By comparing and highlighting the distinction between the two approaches, we aim to provide a clear understanding of their strengths and implications for state-space modelling.

1) ONLINE PREDICTION

Online prediction in the context of SSM refers to making prediction based on the recursive estimation of the system's state. It assesses the models' ability to make one-step-ahead predictions, which is a practical requirement during plant operation. This approach is consistent with the Kalman filter equations

$$x_k^k = x_k^{k-1} + K_k(y_k - Cx_k^{k-1} - Du_k) \tag{28}$$

$$x_{k+1}^k = Ax_k^k + Bu_k \tag{29}$$

$$\hat{y}_{k+1} = Cx_{k+1}^k + Du_{k+1}, \tag{30}$$

where the most recent observation is incorporated and the state estimate is corrected with the prediction error, leading to potentially more accurate and timely predictions.

2) OFFLINE PREDICTION

Offline prediction in this study refers to predicting the future states or outputs without updating the state variables based on new observations. It tests the model's ability to capture long-term trends, dynamics, and fluctuations in the data. This kind of prediction does not employ any correction mechanism from future observations, and its accuracy relies heavily on the proper initialization of the model and its parameters.

$$x_k^k = x_k^{k-1} \tag{31}$$

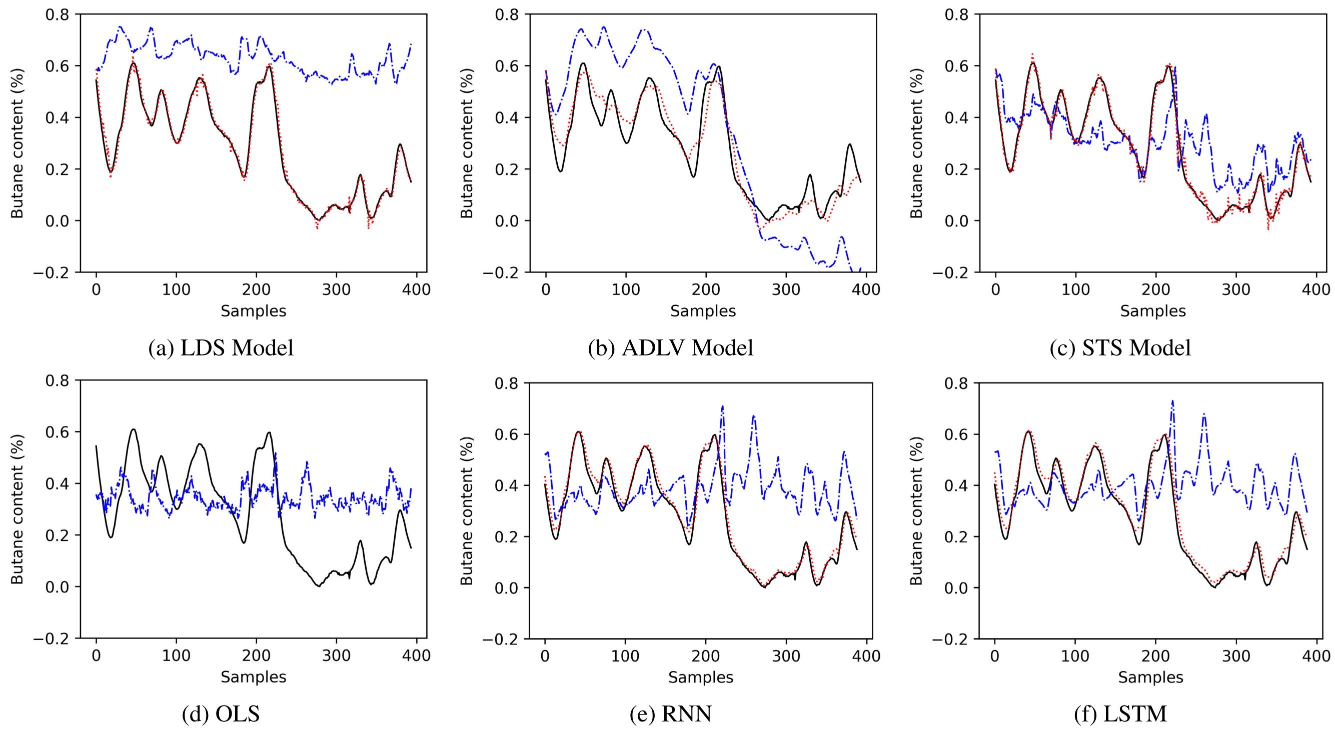
$$x_{k+1}^k = Ax_k^k + Bu_k \tag{32}$$

$$\hat{y}_{k+1} = Cx_{k+1}^k + Du_{k+1} \tag{33}$$

As the equations show, prediction of target variable $\hat{y}_k$ offline requires only the system's model and the known inputs (process data). If the model captures the dynamic of the system correctly and matches the actual measurements in testing phase, it is expected to operate on its own even without the update from the new measurements. This independence of operation helps relieve the burden of constantly monitoring the target through laboratory experiments.

IV. EXPERIMENTS AND RESULTS

The research on the three modeling methods discussed in this paper pointed to the relevancy of a comparative analysis of



(a) LDS Model

(b) ADLV Model

(c) STS Model

(d) OLS

(e) RNN

(f) LSTM

FIGURE 2. Online and/or offline prediction results of the butane content of the six models. Black solid line is the measurements of the test set, red dashed line represents the online prediction results, and the blue dash-dot line denotes the corresponding offline predictions.

TABLE 2. Descriptions and analysis about the case studies.

	Debutanizer Column	Melt Index	Sulfur Recovery
Size	2394	331	10072
Dimension	7	17	20
Noise level	Low	High	High
Process	Desulfuring	Polymerization	Sulfur recovery
Target	By-product	Product	By-product

their practical effects in the soft sensor prediction problem. Such comparison can assist researchers and practitioners design their models when building soft sensors. For this purpose, three benchmarks derived from real industrial applications are employed as case studies. These datasets present different statistical properties and cover representative types of processes and variables of interest, as shown in Table. 2. We chose these datasets to provide a discussion on the effects of the modeling techniques applied to soft sensor prediction.

In addition to comparing the three dynamic models, we also include the ordinary least squares (OLS) regression as the baseline and two representative neural networks, namely RNN and LSTM, in the comparison experiments. After training the models, they are tested in two ways, online prediction and offline prediction, where applicable. To maintain consistency across the experiments, we also evaluate the RNN and LSTM models in an offline mode, where the true observations are substituted with the network's predictions.

To evaluate the prediction performance of the comparison methods used in our case studies, we employ the root mean square error (RMSE) and mean absolute error (MAE) as the evaluation metrics. Next, we describe the performed experiments in detail.

A. DEBUTANIZER COLUMN DATASET

The debutanizer column dataset has been commonly used as a study case for soft sensor development [4], [16], [31]. In a desulfuring and naphtha splitter plant in Italy, 7 sensors, including top and bottom temperature, pressure, reflux flow, etc., are installed to monitor the butane concentration in stabilized gasoline. A total of 2394 data samples have been collected in the process, among which the first 2000 samples are used for model training, while the remaining 394 instances are used for testing.

For RNN and LSTM, the initial 2000 samples are further split, reserving 400 samples for model validation. In the experiments, both RNN and LSTM are composed of one recurrent layer, 10 features in the hidden state, Tanh activation neural networks, and the learning rate and epoch number are determined based on the specific application requirements. And the length of the input sequence is set to be 3.

The initialization for the EM algorithm and Kalman filter plays an important role for its convergence and stability. For all three models, they are performed with approximate diffuse initialization, assuming little knowledge of the initial states, i.e., zeros for initial state μ_0 and $\Sigma_0 = \alpha I$, where $\alpha = 100$ and I is the identity matrix. Matrices A, Q, and R are assumed to be αI with $\alpha = 1, 0.0005$, and 0.1, respectively. For the ADLV model, matrix B was initialized by the right singular vectors of the input data, conveniently obtained through the PCA algorithm, whereas C was initialized randomly, and the dimension of the state d is assumed to be 3. Finally, the H and C matrices in the STS model were also initialized randomly, and the dimension of the state r is set as 2.

TABLE 3. The RMSE/MAE performances for butane content prediction of the comparison methods and the iteration number required.

	Iteration	Training	Test (Online)	Test (Offline)
OLS	1	0.1373/0.0933	-	0.1958/0.1660
LDS	10	0.0109/0.0050	**0.0203/0.0145**	0.3820/0.3498
ADLV	3	0.0416/0.0302	0.0571/0.0451	0.2174/0.1923
STS	20	0.0022/0.0015	0.0221/0.0159	**0.1331/0.1094**
RNN	1000	0.0224/0.0173	0.0245/0.0195	0.2303/0.1884
LSTM	1500	0.0330/0.0230	0.0299/0.0245	0.2346/0.1913

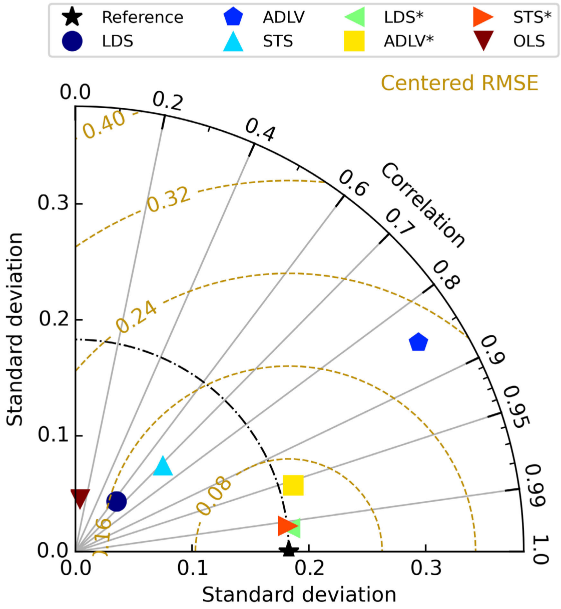


FIGURE 3. Taylor diagram of predictions for debutanizer column dataset produced by different models (Superscript of * to model names refers to results of online predictions, and the next two Taylor diagrams follow the same mark.).

The comparison results of the debutanizer dataset are summarized in Fig. 2, Table 3, and Fig. 3. The measurements in the test set exhibited wavy fluctuations across time and changed smoothly. In the predictions, the baseline model, OLS, only captured the pattern locally with many undesired small variations. For the other methods, they all produced accurate predictions for online predictions, whereas the offline predictions exhibited different patterns.

The predictions made by the LDS model drifted directly from observations when no labels were available, indicating a disagreement between the learned model and the *true* model. Comparatively, the ADLV model and STS model generalized better than the LDS model, as shown in Table 3. The ADLV model is distinguished by its smooth predictions that matched the pattern of the measurement in general, and the predictions results of STS model had the lowest RMSE. Furthermore, the predictions of the RNN and the LSTM models were similar to that of the OLS model and drifted away from the observations in the middle.

We also plotted Taylor diagram [32] as a goodness-of-fit measure to evaluate the performance of the compared models. Taylor diagrams summarize three performance metrics of each model in the prediction of one dataset in a single plot: standard deviation of the predictions indicated by the distance to the origin, centered RMSE (CRMSE) shown in contours, and Pearson correlation coefficient between the observed and the simulated related to the azimuthal angle.

In Fig. 3, the online predictions of the three models all demonstrate similarities to the observations (black star on the x-axis), i.e., closer standard deviations and higher correlations to the reference point. Conversely, the predictions in the offline mode are marked by larger standard deviation (the ADLV model) and lower correlations to the observations (the LDS and STS models). It is noted that the ADLV model has the least change of performance between online and offline predictions in terms of correlation, demonstrating reliability compared to the other two models.

B. MELT INDEX DATASET

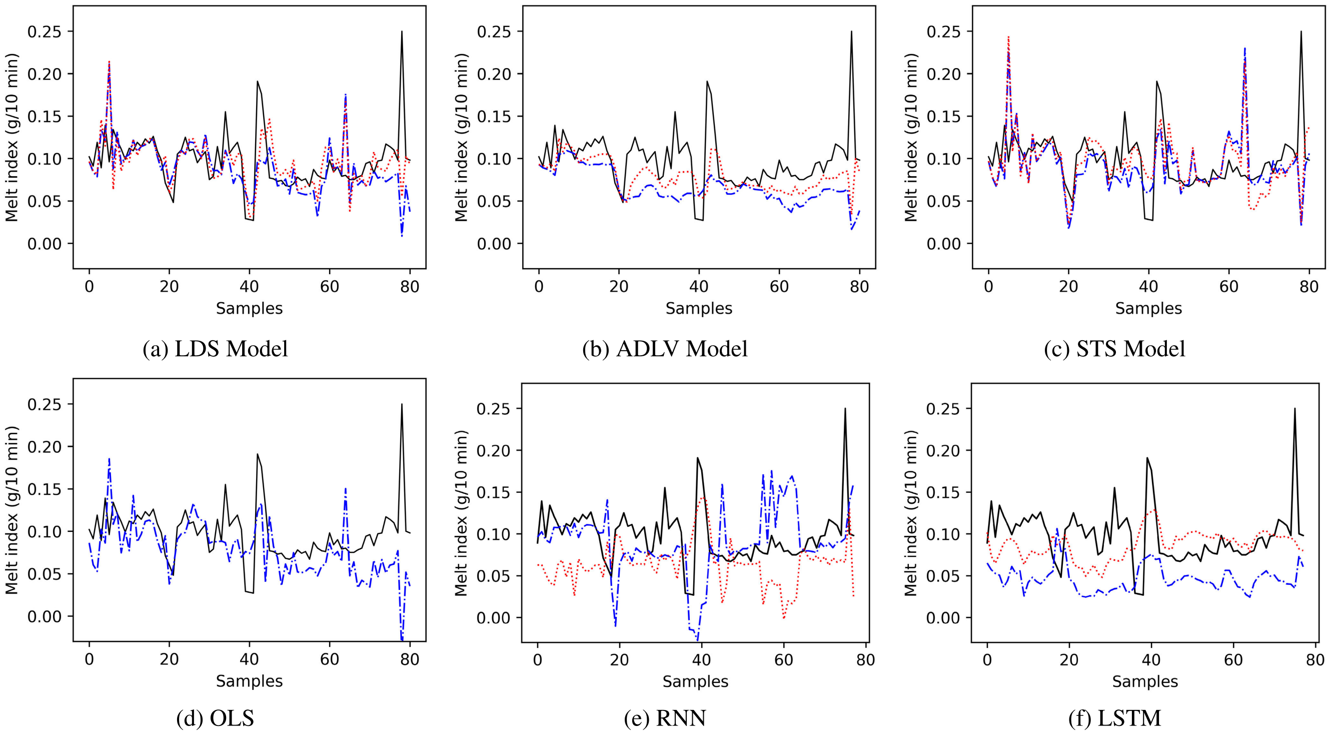
The second dataset was collected from the daily records of process data and laboratory measurements in an industrial polyethylene process plant in Taiwan [33]. After preprocessing, 17 features from the second and third reactors are selected as input data, while the melt index of the third reactor serves as the output variable for the soft sensor. Among the 331 observations, those collected from July 2009 (the first 249 samples) are used for model training, and the data acquired since 2011 are used for testing. For the RNN and LSTM, the last 49 samples of the training set are used for validation.

The initialization for the EM algorithm and the Kalman filter remained in a similar way as in the last experiment, except the initialization of the state covariance matrix $Q = \alpha I$ for the ADLV model and the STS model, with $\alpha = 0.1$ and $\alpha = 0.01$, respectively, and an initialization of smaller observation error variance, with $R = 0.05$.

TABLE 4. The RMSE/MAE performances for melt index prediction of the comparison methods and the iteration number required.

	Iteration	Training	Test (Online)	Test (Offline)
OLS	1	0.0693/0.0406	-	0.0487/0.0328
LDS	5	0.0386/0.0252	**0.0360/0.0213**	**0.0398/0.0242**
ADLV	6	0.0484/0.0290	0.0377/0.0236	0.0473/0.0347
STS	5	0.0582/0.0367	0.0429/0.0272	0.0428/0.0262
RNN	1500	0.0934/0.0601	0.0501/0.0412	0.0540/0.0354
LSTM	1500	0.0987/0.0664	0.0392/0.0301	0.0632/0.0553

The comparison results for the melt index dataset are summarized in Fig. 4, Fig. 5 and Table 4, respectively. Compared to the previous example, the target variable in this case study exhibited more drastic fluctuations over time. The



(a) LDS Model (b) ADLV Model (c) STS Model

(d) OLS (e) RNN (f) LSTM

FIGURE 4. Online and/or offline prediction results of the melt index of the six models. Black solid line is the measurements of the test set, red dashed line represents the online prediction results, and the blue dash-dot line denotes the corresponding offline predictions.

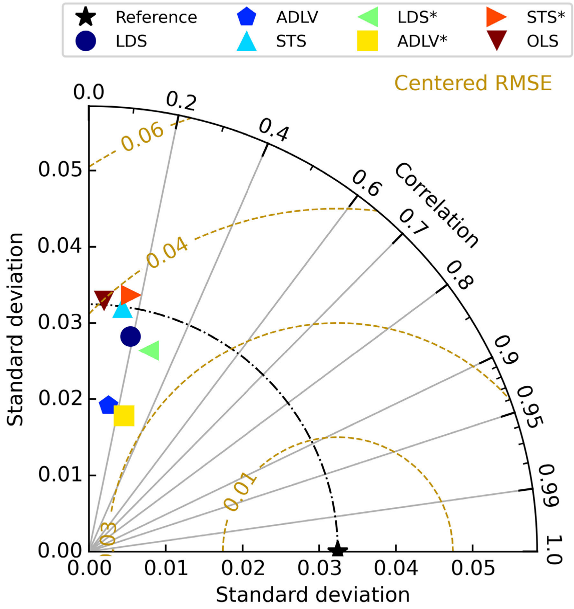


FIGURE 5. Taylor diagram of predictions for melt index dataset produced by different models.

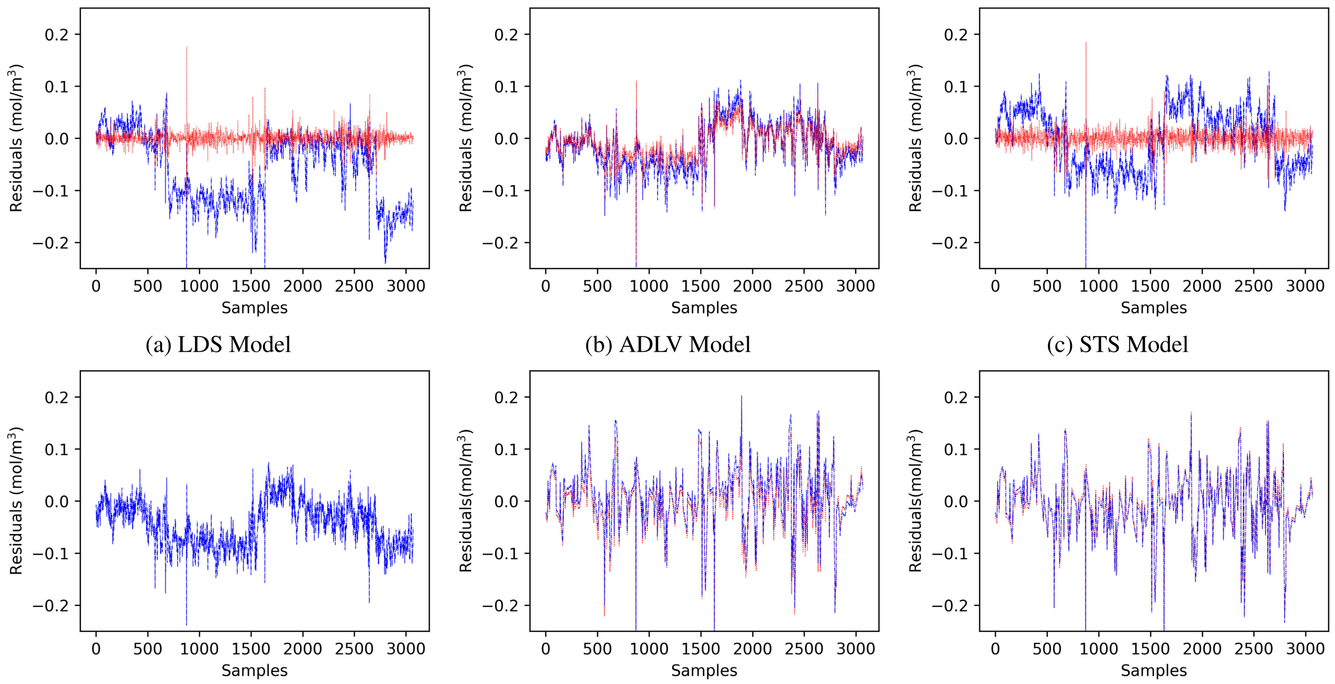
OLS regression model with 17 features achieved partially satisfactory prediction performance. It predicted well for the first 60 samples but gradually deviated from the true pattern in the latter part, possibly due to its inability to capture new system dynamics.

In this example, the three methods had similar prediction errors and lower correlations to the observed in general, which makes the selection of the best model more difficult. Although the offline prediction results of the ADLV model were closer to the observed, the results of the STS model presented a similar standard deviation in scale, and the results of the LDS model were more balanced in these two aspects. Moreover, the distances between the online predictions and offline predictions of the three models were relatively close, showing in three pairs in Fig. 5. This suggests that all three models produced reasonable results and model selection depends on considerations for different aspects of predictions.

Regarding the results of RNN and LSTM, their complex structures did not benefit this task and yielded unsatisfactory results. As shown in Fig. 4e, the RNN model produced predictions in opposite directions and showed latency in its predictions, particularly near the end of the test set. On the other hand, the prediction results of LSTM are flatter and suggest a gap in predictions after switching to offline mode. Nevertheless, training neural networks required significant effort in parameter tuning compared to the state-space models, and the resulted models lacked interpretation.

C. SULFUR RECOVERY UNIT DATASET

The third case study is based on another real-life industrial chemical process, specifically the sulfur recovery unit (SRU) process, which aims to remove environmental pollutants from



(a) LDS Model

(b) ADLV Model

(c) STS Model

(d) OLS

(e) RNN

(f) LSTM

FIGURE 6. **Online and/or offline prediction results of the SO_2 concentration of the six models. Red dashed line represents the online prediction residuals, and the blue dash-dot line denotes the corresponding offline prediction residuals.**

acid gases. The SRU dataset is a benchmark dataset for soft sensor design [11], [31], which takes five different acid gas flows as input to predict the concentrations of hydrogen sulfide (H_2S) and sulfur dioxide (SO_2) in the tail stream. Based on prior expert analysis of the process dynamics, the target variable is considered related not only to the current input data but also to the input query sampled 5, 7, and 9 time steps earlier [31]. This leads to a total of 20 dimensions after appending the input data.

For this study, we choose the concentration of sulfur dioxide (SO_2) as the target variable. After pre-processing, the SRU dataset contains 10,072 samples of 20 process variables and one target variable. Among which, the first 7000 samples are used as the historical training data for OLS and the three dynamic models. For the RNN and LSTM, the latter 2000 samples of the training set are used for model validation. The remaining 3072 samples form the test set.

For this case study, the EM algorithm and Kalman filter were initialized with $R = 0.01$, and $Q = \alpha I$, where $\alpha = 0.0001$ for the LDS model and $\alpha = 0.01$ for the ADLV and STS models, and the transition matrix $A = 0.1I$ for the STS model. All the other parameters were generated consistently as the first two examples. For the training of RNN and LSTM, two-layer neural networks with 100 hidden nodes were employed and trained for 250 epochs.

Table 5 summarizes the prediction results of the comparison methods on the test set of the third case study. Although the offline prediction results of all three models present similar patters in Fig. 6, according to Fig. 7, however, the result of the ADLV model was closer to the observed, had similar standard deviation and smaller RMSE than the other two methods. It is also noted that the ADLV model had the least changes in all three metrics between the online predictions and offline predictions, as shown in Fig. 6b. The consistency of performance is a desired quality when applied in practical applications. Lastly, both the online and offline predictions of RNN and LSTM appeared very flat and lacked details, resulting in large residuals compared to the other models. This suggests that the RNN and LSTM models might not be as suitable for this particular application.

TABLE 5. **The RMSE/MAE performances for SO_2 concentration prediction of the comparison methods and the iteration number required.**

	Iteration	Training	Test (Online)	Test (Offline)
OLS	1	0.0330/0.0209	-	0.0618/0.0507
LDS	5	0.0047/0.0029	**0.0132/0.0084**	0.0899/0.0698
ADLV	3	0.0251/0.0151	0.0314/0.0257	**0.0466/0.0386**
STS	10	0.0100/0.0055	0.0158/0.0108	0.0581/0.0509
RNN	250	0.0595/0.0361	0.0522/0.0382	0.0560/0.0413
LSTM	250	0.0607/0.0372	0.0543/0.0394	0.0547/0.0400

V. DISCUSSION

Our experiment results suggest that the prediction performance of each dynamic models for for soft sensor problems were dependent on the characteristics of the dataset and the nature of the process. And the significance of choosing a suitable modelling method is visualized in Fig. 8,

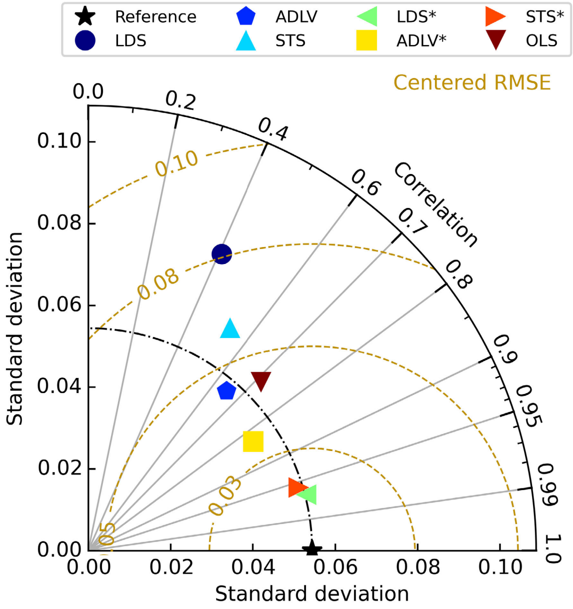


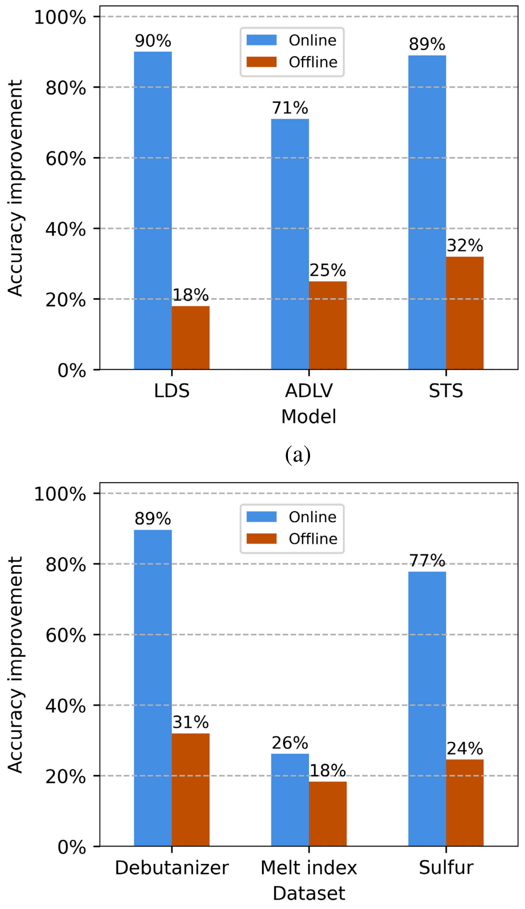
FIGURE 7. Taylor diagram of predictions for sulfur dataset produced by different models.

which shows the observed maximum prediction accuracy improvements (measured by the decrease in RMSE error) over the OLS regression approach.

As shown in Fig. 8, the online approach consistently shows a significantly higher accuracy improvement compared to the offline approach. This suggests that the online methods are more effective at adjusting to changes in real-time data. It can be observed that the LDS model and the STS model show higher online accuracy improvements (90% and 89%), demonstrating good adaptability and flexibility, particularly when the observations are assumed to be measured with high resolution. This may also relate to their having less parameters compared to the ADLV model and are therefore easier to train.

Furthermore, by analyzing the decrease in prediction accuracy across the online and offline predictions for each model in Fig. 8, we observe that the ADLV model is the most consistent among the three models. This observation is in agreement with the results in the prediction plots and Taylor diagrams shown earlier, which suggests that the complexity of the ADLV model might allow it to capture more nuanced patterns through dimensionality reduction. Given this results, the ADLV model may be the preferred choice if an application requires offline predictions.

Moreover, our experiment results also indicate dependence upon the characteristic nature of the datasets. We believe that understanding the difference in the intrinsic processes helps selection of a suitable model. Accordingly, Fig. 8b presents this complementary information. It can be observed that the dependence on datasets is stronger than that of models. The debutanizer column dataset has the highest online prediction improvement at 89% and most offline prediction improvement at 31%. The results of the SRU dataset follow a similar pattern, while the melt index dataset exhibits the least online (26%) and offline (18%) improvements. Next, we analyze this results from two aspects, the data perspective and the process perspective.



(a)

(b)

FIGURE 8. Maximum online and offline prediction accuracy improvements. (a) presents model-wise improvement across three datasets. (b) presents dataset-wise improvement between three models.

According to Table. 2, the debutanizer column dataset and the SRU dataset share many similarities. Their larger sizes compared to the dimensions allow for the successful training of more complex models, such as the ADLV model. Conversely, datasets with smaller sizes and larger dimensions, such as the melt index dataset, may cause overfitting problems. These differences in data statistics are also related to sampling frequencies. Despite the large number of samples in the SRU dataset, its high sampling frequency (every minute) only covers a short period of time (about 7 days). In contrast, the melt index dataset was sampled daily and spanned almost a whole year. The smaller size and longer time span make it more suitable for a time-varying coefficients model like the LDS model.

Differences in the nature of the process and the variable of interest also help explain the performance of each model.

The polymerization process focuses on the ease of the flow of the melt of the produced material and is directly related to the quality (average molecular weight) of the end product. In contrast, the desulfuring and the sulfur recovery processes concentrate on the by-products (butane content and hydrogen sulfide concentration) and aim to reduce environmental pollutants and contaminants. In this context, the pollutants need to be minimized, and the models are designed to signal abnormal high-level concentrations to optimize their removal. The complex chemical reactions involved and the indirect links between the process data and the key variable may require more complex models (like the ADLV and STS models) to capture the underlying dynamics.

VI. CONCLUSION

To sum up, this paper presents three representative state-space formulations for dynamic soft sensing applications. These different models need to be selected appropriately to capture the underlying dynamics of the process and ensure prediction accuracy. For this purpose, we evaluated three modelling methods commonly used in real-world applications. We aim to contribute to the process of formulation, selection and application of SSM-based dynamic models encountered in industrial projects.

A detailed assessment of the effects of these three models on soft sensor predictions was conducted based on our experiment results. The high percentage improvements indicate that all three models are significantly better than the baseline OLS approach when implemented online. Specifically, we observed that the results of the LDS model were consistently the best among all models for online predictions. The results of the STS model were in similar range to that of the LDS model. The online approach's superiority suggests that it's better suited for adapting to dynamic environments, but it may also require continuous data streams and could be more sensitive to data anomalies or noise.

On the other hand, the offline predictions produced by the three models demonstrated varying patterns and there was no best model across the three datasets. Factors including data size, sampling frequency, the intrinsic characteristics of the process all influence the prediction results. Nonetheless, it was possible for us to observe that the ADLV model produced the most consistent predictions across online and offline predictions, which indicates its ability to work with larger datasets and potentially complex relationships. In practice, this means that the model is reliable even when it is not updated by the laboratory measurements. We also note that the result of the STS model was closely related to that of the OLS model and can generally improve its result. Understanding these effects helps guide the selection of appropriate models for different applications and data types for soft sensor development in industrial processes.

It should be noted that modelling and interpreting the underlying dynamics of the physical process based on the collected process data is a very challenging problem. Real-world data may present more complex nonstationarity and show varying characteristics across applications. The models we discuss in this paper lay a foundation on the subject upon which further theoretical exploration of more sophisticated predictive frameworks can be constructed. We also suggest future research to validate the models across different domains and industries.

APPENDIX

$$S_{xx} = \sum_{k=1}^{n} (P_k^n + x_k^n (x_k^n)^T) \tag{34}$$

$$S_{xb} = \sum_{k=1}^{n} (P_{k,k-1}^n + x_k^n (x_{k-1}^n)^T), \quad S_{bx} = S_{xb}^T \tag{35}$$

$$S_{bb} = \sum_{k=1}^{n} (P_{k-1}^n + x_{k-1}^n (x_{k-1}^n)^T) \tag{36}$$

$$S_{xz} = \sum_{k=1}^{n} x_k^n u_{k-1}^T, \quad S_{zx} = S_{xz}^T \tag{37}$$

$$S_{bz} = \sum_{k=1}^{n} x_{k-1}^n u_{k-1}^T, \quad S_{zb} = S_{bz}^T \tag{38}$$

$$S_{zz} = \sum_{k=1}^{n} u_{k-1} u_{k-1}^T \tag{39}$$

$$S_{yy} = \sum_{k=1}^{n} y_k y_k^T \tag{40}$$

$$S_{yx} = \sum_{k=1}^{n} y_k (x_k^n)^T, \quad S_{xy} = S_{yx}^T \tag{41}$$

$$S_{yu} = \sum_{k=1}^{n} y_k u_k^T, \quad S_{uy} = S_{yu}^T \tag{42}$$

$$S_{xu} = \sum_{k=1}^{n} x_k^n u_k^T, \quad S_{ux} = S_{xu}^T \tag{43}$$

$$S_{uu} = \sum_{k=1}^{n} u_k u_k^T \tag{44}$$

$$A^{\text{new}} = (S_{xb} - S_{xz} S_{zz}^{-1} S_{zb})(S_{bb} - S_{bz} S_{zz}^{-1} S_{zb})^{-1} \tag{45}$$

$$B^{\text{new}} = (S_{xz} - S_{xb} S_{bb}^{-1} S_{bz})(S_{zz} - S_{zb} S_{bb}^{-1} S_{bz})^{-1} \tag{46}$$

$$C^{\text{new}} = (S_{yx} - S_{yu} S_{uu}^{-1} S_{ux})(S_{xx} - S_{xu} S_{uu}^{-1} S_{ux})^{-1} \tag{47}$$

$$D^{\text{new}} = (S_{yu} - S_{yx} S_{xx}^{-1} S_{xu})(S_{uu} - S_{ux} S_{xx}^{-1} S_{xu})^{-1} \tag{48}$$

ACKNOWLEDGMENT

The authors are grateful to ENEOS Corporation for providing a motivation and many valuable comments for this study.

REFERENCES

[1] M. West, P. J. Harrison, and H. S. Migon, "Dynamic generalized linear models and Bayesian forecasting," *J. Amer. Stat. Assoc.*, vol. 80, no. 389, pp. 73–83, Mar. 1985.

[2] V. K. Puli and B. Huang, "Variational Bayesian approach to nonstationary and oscillatory slow feature analysis with applications in soft sensing and process monitoring," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 4, pp. 1708–1719, Jul. 2023.

[3] J. Gama, I. Žliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surveys*, vol. 46, no. 4, pp. 1–37, Mar. 2014.

[4] Y. Jiang, S. Yin, J. Dong, and O. Kaynak, "A review on soft sensors for monitoring, control, and optimization of industrial processes," *IEEE Sensors J.*, vol. 21, no. 11, pp. 12868–12881, Jun. 2021.

[5] M. Sugiyama, S. Nakajima, H. Kashima, P. Buenau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS) 2007,* Vancouver, BC, Canada, Dec. 2007.

[6] F. Bayram, B. S. Ahmed, and A. Kassler, "From concept drift to model degradation: An overview on performance-aware drift detectors," *Knowl.-Based Syst.*, vol. 245, Jun. 2022, Art. no. 108632.

[7] A. Torgashov and S. Skogestad, "The use of first principles model for evaluation of adaptive soft sensor for multicomponent distillation unit," *Chem. Eng. Res. Des.*, vol. 151, pp. 70–78, Nov. 2019.

[8] R. Zhang, A. Xue, and F. Gao, "Temperature control of industrial coke furnace using novel state space model predictive control," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2084–2092, Nov. 2014.

[9] W. Favoreel, B. De Moor, and P. Van Overschee, "Subspace state space system identification for industrial processes," *IFAC Proc. Volumes*, vol. 31, no. 11, pp. 319–327, Jun. 1998.

[10] C. M. Bishop and N. M. Nasrabadi, "Sequential data," in *Pattern Recognition and Machine Learning*, 1st ed. New York, NY, USA: Springer, 2006, pp. 642–643.

[11] B. Bidar, F. Shahraki, J. Sadeghi, and M. M. Khalilipour, "Soft sensor modeling based on multi-state-dependent parameter models and application for quality monitoring in industrial sulfur recovery process," *IEEE Sensors J.*, vol. 18, no. 11, pp. 4583–4591, Mar. 2018.

[12] P. Cao and X. Luo, "Modeling for soft sensor systems and parameters updating online," *J. Process Control*, vol. 24, no. 6, pp. 975–990, Jun. 2014.

[13] T. Xu, Y. Chen, D. Zeng, and Y. Wang, "Mixed-response state-space model for analyzing multi-dimensional digital phenotypes," *J. Amer. Stat. Assoc.*, vol. 119, pp. 1–23, Jul. 2023.

[14] F. T. Dastjerd, J. Sadeghi, F. Shahraki, M. M. Khalilipour, and B. Bidar, "Soft sensor design using multi-state dependent parameter methodology based on generalized random walk method," *IEEE Sensors J.*, vol. 22, no. 8, pp. 7888–7901, Apr. 2022.

[15] G. Liang, S. Ren, and F. Dong, "Dynamic imaging for time-varying distribution using electrical/ultrasonic dual-modality tomography," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–13, 2022.

[16] X. Yuan, Y. Wang, C. Yang, Z. Ge, Z. Song, and W. Gui, "Weighted linear dynamic system for feature representation and soft sensor application in nonlinear dynamic industrial processes," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1508–1517, Feb. 2018.

[17] L. Wiskott and T. J. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural Comput.*, vol. 14, no. 4, pp. 715–770, Apr. 2002.

[18] Q. Wen, Z. Ge, and Z. Song, "Data-based linear Gaussian state-space model for dynamic process monitoring," *AIChE J.*, vol. 58, no. 12, pp. 3763–3776, Dec. 2012.

[19] G. Li, B. Liu, S. J. Qin, and D. Zhou, "Quality relevant data-driven modeling and monitoring of multivariate dynamic processes: The dynamic T-PLS approach," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2262–2271, Dec. 2011.

[20] Y. Ma and B. Huang, "Bayesian learning for dynamic feature extraction with application in soft sensing," *IEEE Trans. Ind. Electron.*, vol. 64, no. 9, pp. 7171–7180, Sep. 2017.

[21] S. J. Qin, Y. Dong, Q. Zhu, J. Wang, and Q. Liu, "Bridging systems theory and data science: A unifying review of dynamic latent variable analytics and process monitoring," *Annu. Rev. Control*, vol. 50, pp. 29–48, Jan. 2020.

[22] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 6645–6649.

[23] L. R. Medsker and L. C. Jain, "Efficient second-order learning algorithms for discrete-time recurrent neural networks," in *Recurrent Neural Networks. Design and Applications*, 1st ed. New York, NY, USA: CRC Press, Dec. 2001, pp. 65–67.

[24] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Atlanta, GA, USA, Feb. 2013, pp. 1310–1318.

[25] J. Durbin and S. J. Koopman, "Linear state space models," in *Time Series Analysis by State Space Methods*, vol. 38, 2nd ed. Oxford, U.K.: OUP Oxford, 2012, pp. 43–75.

[26] R. Salles, K. Belloze, F. Porto, P. H. Gonzalez, and E. Ogasawara, "Nonstationary time series transformation methods: An experimental review," *Knowl. Based Syst.*, vol. 164, pp. 274–291, Jan. 2019.

[27] A. Stathopoulos and M. G. Karlaftis, "A multivariate state space approach for urban traffic flow modeling and prediction," *Transp. Res. C, Emerg. Technol.*, vol. 11, no. 2, pp. 121–135, Apr. 2003.

[28] S. Liu and A. Lehrmann, "DynaConF: Dynamic forecasting of non-stationary time-series," 2022, *arXiv:2209.08411*.

[29] L. Zhou, G. Li, Z. Song, and S. J. Qin, "Autoregressive dynamic latent variable models for process monitoring," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 1, pp. 366–373, Jan. 2017.

[30] K. Chen, I. Castillo, L. H. Chiang, and J. Yu, "Soft sensor model maintenance: A case study in industrial processes," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 427–432, Jan. 2015.

[31] L. Fortuna, S. Graziani, A. Rizzo, and X. M. Gabriella, "Appendix A: Description of the plants," in *Soft Sensors for Monitoring and Control of Industrial Processes*, 1st ed. London, U.K.: Springer, May 2007, pp. 229–231.

[32] K. E. Taylor, "Summarizing multiple aspects of model performance in a single diagram," *J. Geophys. Res., Atmos.*, vol. 106, pp. 7183–7192, Apr. 2001.

[33] Y. Liu, Z. Gao, and J. Chen, "Development of soft-sensors for online quality prediction of sequential-reactor-multi-grade industrial processes," *Chem. Eng. Sci.*, vol. 102, pp. 602–612, Oct. 2013.

WENYI LIU was born in Henan, China, in 1995. She received the B.S. degree in automation and the M.S. degree in control science and engineering from Xi'an Jiaotong University, Shaanxi, China, in 2017 and 2020, respectively. She is currently pursuing the Ph.D. degree with the Department of Advanced Interdisciplinary Studies, The University of Tokyo, Tokyo, Japan. Her research interests include soft sensor development, time series modelling, and data drift problem.

TAKEHISA YAIRI (Member, IEEE) received the B.Eng., M.Sc., and Ph.D. degrees in aerospace engineering from The University of Tokyo, Japan, in 1994, 1996, and 1999, respectively. He is currently a Professor with the Research Center for Advanced Science and Technology (RCAST), The University of Tokyo. His research interests include anomaly detection, health monitoring, fault diagnosis, learning dynamical systems, nonlinear dimensionality reduction, and the applications of machine learning and probabilistic inference to aerospace systems.