Time-Variant Digital Twin Modeling through the Kalman-Generalized Sparse Identification of Nonlinear Dynamics

Jingyi Wang, Jesús Moreira, Yankai Cao, Bhushan Gopaluni

***Abstract*— A digital twin is a computer-based digital representation that simulates the behavior of a physical system. Digital twins help users to interact with real-world processes digitally. Time-variant modeling is critical to preserving the accuracy of digital twin models as the process dynamics change with time. Kalman filter is a well-known recursive algorithm that adjusts the process state estimates using real-time measurements. Sparse identification of nonlinear dynamics (SINDy) is an algorithm that automatically identifies system models from large data sets using sparse regression so as to prevent overfitting and find an ideal trade-off between model complexity and accuracy. In this paper, the SINDy approach is first extended to the generalized SINDy (GSINDy). Then, the GSINDy is integrated with Kalman filter to automatically identify time-variant digital twin models for online applications. The effectiveness of the algorithm is revealed through a simulation example based on Lorenz system and an industrial diesel hydrotreating unit example.**

I. Introduction

Modern industrial processes are equipped with hundreds of sensors that generate large volumes of data [1], [2]. The data from these sensors enable us to significantly improve productivity as well as economic and environmental efficiency. Digital twins are models that are developed using large volumes of real-time data to simulate industrial processes, which are central to the concept of Industry 4.0 in smart manufacturing [3]. A digital twin captures information, such as data records, asset tags, sensor conditions, etc., from the physical assets [4]. Accordingly, a digital twin will provide predictions and insights, which can be used to make decisions on the operational conditions of the physical system [5]. A graphical illustration of typical interactions between physical assets and digital twins is shown in Fig. 1.

The accuracy and robustness of digital twin models are critical to bridging the gap between virtual design and real-life manufacturing [6]. Since a large-scale digital twin may contain various types of information and need to provide predictions for different operational objectives, a multi-input, multi-output (MIMO) modeling approach is desired [7]. In [6], a comprehensive reference digital twin modeling approach was proposed to address the scalability, interoperability, and fidelity of a digital twin. Automation machine learning was applied for digital twin attributes modeling in [8] to improve the effectiveness of data exchange among various systems in a digital twin. In [9], the idea of five-dimension architecture of digital twin is proposed, including physical entity, virtual model, connection, digital twin data, and services. However, the automatic, time-variant digital twin modeling approach has not been well studied.

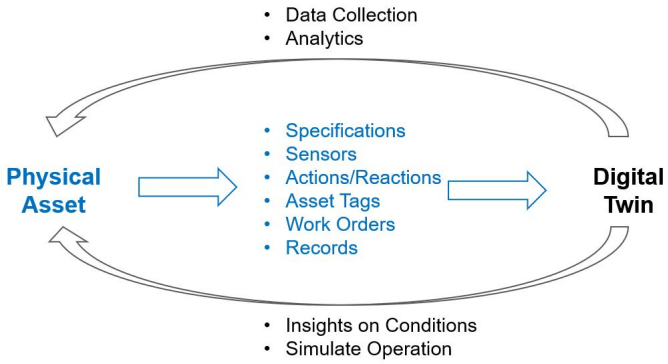

Fig. 1. Relationship between the physical assets and digital twins.

Traditionally, it takes years for scientists to develop first-principle models describing process dynamics. In recent years, with rapid development of artificial intelligence tools, data-driven modeling methods are playing a critical role in modeling problems [10], [11]. The sparse identification of nonlinear dynamics (SINDy) proposed by Steven et al. is able to automatically discover the governing equations of dynamic systems from abundant data by solving a sparse regression problem [12]. There exist three main steps in the SINDy[1] - data collection, identification of model term library, and solving a sparse regression problem. By solving the sparse regression problem, the SINDy provides parsimonious models that balance accuracy with model complexity [13].

However, the SINDy requires adequate data samples, and the identified model accuracy is sensitive to noise in data. In [14], a deep learning-based, signal-noise decomposition approach is proposed to separate the noise from noisy measurements using time-stepping constraints. The idea from [14] was employed in [15] to develop the modified SINDy, which has higher robustness to noisy measurements than the SINDy. The modified SINDy requires adequate time-series data, and it also needs to perform an additional estimation of a vector field. In addition, both the SINDy and modified SINDy are limited to time-invariant model constructions. In

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

Jingyi Wang, Yankai Cao, and R. Bhushan Gopaluni are with the Department of Chemical and Biological Engineering, University of British Columbia, 2360 E Mall, Vancouver, BC, V6T 1Z3, Canada, e-mail: wjy1129@student.ubc.ca; yankai.cao@ubc.ca; bhushan.gopaluni@ubc.ca

Jesús Moreira is with the Imperial Oil Company, 505 Quarry Park Blvd SE, Calgary, AB, T2C 5N1, Canada, e-mail: jesus.moreira@esso.ca

[1]For simplicity, we will refer to SINDy algorithm simply as SINDy

[16], a time-varying SINDy framework was developed to apply the SINDy in a moving time window at each time instant. However, the SINDy algorithm itself is still time-invariant.

The Kalman filter and its variants have wide-ranging applications in state estimation, which modify the predictions according to real-time output measurements [17], [18]. In this paper, the SINDy algorithm is first extended to the generalized SINDy (GSINDy), which extends the applicable range of SINDy to general MIMO modelings. Then, the GSINDy is integrated with the Kalman filter to form the Kalman-GSINDy algorithm, which recursively identifies the active digital twin model terms from the SINDy model library. The parameters corresponding to the active model terms are then estimated through an iterative correction procedure to promote sparsity. Consequently, a time-variant digital twin modeling approach is proposed, resulting in time-varying sparse digital twin models that describe the system dynamics.

The remainder of this paper is organized as follows. Section II extends the SINDy to generalized SINDy (GSINDy) to solve general MIMO modeling problems. Section III introduces the Kalman-GSINDy algorithm in detail. One Lorenz system numerical example and one industrial diesel hydrotreating (DHT) unit case study are analyzed in Section IV to demonstrate the advantages of the proposed Kalman-GSINDy time-variant digital twin modeling approach. Finally, section V concludes this paper.

II. The SINDy Adaptation to General Modeling Problems

As mentioned in Section I, the SINDy algorithm is able to automatically discover the sparse governing equations of dynamic systems from large data sets. It includes three main steps, data collection, identification of model term library, and solving a sparse regression problem [12]. However, current SINDy is limited to modeling of time-series relationships, such as ordinary differential equations (ODE) and partial differential equations (PDE) problems, in which the number of input variables and output variables is the same. In this section, the main steps of SINDy are introduced with modifications to extend it to the GSINDy, which is able to solve general MIMO modeling problems.

The first main step in the SINDy is data collection, in which the sensor measurements are collected along time,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_{t_1} & \mathbf{x}_{t_2} & \dots & \mathbf{x}_{t_m} \end{bmatrix}^T, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$, and m represents the number of measurement samples along time. When identifying the ODE and PDE relationships, the derivative of sensor measurement, $\dot{\mathbf{X}}$, is required to be measured or calculated through numerical methods, such as the central difference,

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}_{t_1} & \dot{\mathbf{x}}_{t_2} & \dots & \dot{\mathbf{x}}_{t_m} \end{bmatrix}^T. \tag{2}$$

The GSINDy is no longer limited to time-series dynamics modeling. When identifying the relationship between sensor measurement, $\mathbf{X}$, and system output, $\mathbf{Y}$, the output measurements are collected as a function of time,

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_{t_1} & \mathbf{y}_{t_2} & \dots & \mathbf{y}_{t_m} \end{bmatrix}^T, \tag{3}$$

where $\mathbf{y} \in \mathbb{R}^j$.

The second main step is to construct a library that contains all potential model terms. This library can be highly flexible. It can be a combination of potential first-principle terms and data-driven terms. In applications of the SINDy algorithm, various types of SINDy libraries have been developed, including polynomial, Fourier, trigonometric libraries, etc. In addition, these libraries can be combined together to form a new library. A sample combined polynomial and trigonometric library is

$$\boldsymbol{\Theta}(\mathbf{X}) = \begin{bmatrix} \mathbf{1} & \mathbf{X} & \mathbf{X}^{P_2} & \dots & \sin(\mathbf{X}) & \cos(\mathbf{X}) & \dots \end{bmatrix}, \tag{4}$$

where $\mathbf{X}^{P_i}$ represents all possible combinations in an i^{th}-order polynomial. For example,

$$\mathbf{X}^{P_2} = \begin{bmatrix} x_{1;t_1}^2 & x_{1;t_1}x_{2;t_1} & \dots & x_{2;t_1}^2 & \dots & x_{n_x;t_1}^2 \\ x_{1;t_2}^2 & x_{1;t_2}x_{2;t_2} & \dots & x_{2;t_2}^2 & \dots & x_{n_x;t_2}^2 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{1;t_m}^2 & x_{1;t_m}x_{2;t_m} & \dots & x_{2;t_m}^2 & \dots & x_{n_x;t_m}^2 \end{bmatrix}. \tag{5}$$

After constructing the candidate model term library, the sparse regression problem between the output, $\mathbf{Y}$, and candidate model terms, $\boldsymbol{\Theta}(\mathbf{X})$, can be solved using the following equation,

$$\mathbf{Y} = \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\Xi}, \tag{6}$$

where each column inside $\boldsymbol{\Xi}$, is ξ_k, the determining parameters of active terms in k^{th} column of $\boldsymbol{\Theta}(\mathbf{X})$. By solving the sparse regression problem, we force most entries in $\boldsymbol{\Xi}$ to zero and thus promote the sparsity of identified models. In this case, the identified process model for one of the objective outputs can be written as:

$$y_k = \boldsymbol{\Theta}(\mathbf{x})\xi_k, \tag{7}$$

where ξ_k is the model parameter vector for the k^{th} objective output, y_k, and $\boldsymbol{\Theta}(\mathbf{x})$ is a row vector containing symbolic functions of $\mathbf{x}$, which is different from $\boldsymbol{\Theta}(\mathbf{X})$, referring to a data matrix [12].

Fig. 2 illustrates a sample GSINDy result for a MIMO modeling problem. In this example, two output functions are aimed to be identified, and then a third-order polynomial library about the input, $\mathbf{x}$, is constructed, where $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$. After solving the sparse regression problem, the identified active model terms in $\boldsymbol{\Xi}$ are marked using the blue and yellow dots for y_1 and y_2, respectively. The result shows that two active model terms are selected from the library, $\boldsymbol{\Theta}(\mathbf{x})$, for y_1, and three active model terms are selected for y_2, respectively.

Several approaches can be used to solve the sparse regression problem. One of the famous methods is the least absolute shrinkage and selection operator (LASSO), which adds penalization terms to the optimization equation of least

square regression to promote sparsity. However, LASSO will produce parameters with small magnitudes, but not exactly zero magnitude, and hence it does not guarantee to result in sparse models. The sequential least squares (SLS) algorithm promotes model sparsity through an iterative procedure. In the SLS, a parameter magnitude threshold, λ, is set, and then the resulting parameters whose absolute values are smaller than λ are forced to be equal to zero. Afterward, another regression and thresholding procedure is performed on the nonzero terms. The procedure is repeated until convergence [19].

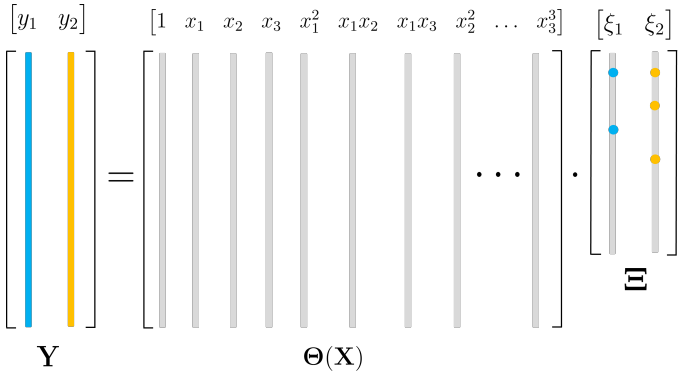

Fig. 2. Graphical illustration of the GSINDy result for a MIMO modeling problem.

III. The Kalman-GSINDy Algorithm

In Section II, we developed the GSINDy algorithm that is applicable to general MIMO modeling problems. In this section, the Kalman filter algorithm is first introduced, and then it is integrated with the GSINDy algorithm to recursively select the active model terms and estimate the corresponding model parameters. The Kalman filter and GSINDy integrated algorithm is referred as the Kalman-GSINDy.

A. The Kalman Filter Algorithm

Consider a linear system:

$$x_t = F_t x_{t-1} + w_{t-1}, \tag{8}$$

$$y_t = H_t x_t + v_t, \tag{9}$$

where F_t and H_t represent the linear state and output models, respectively; x_t is the state, and y_t denotes the output measurement. Process and measurement noises are denoted as w_t and v_t, which are assumed to be zero-mean, uncorrelated, Gaussian white noise. Their corresponding covariance matrices are Q and R, respectively.

Accordingly, the Kalman filter algorithm can be summarized as follows [20].

Prediction:

$$\hat{x}_t^- = F_t \hat{x}_{t-1}, \tag{10}$$

$$P_t^- = F_{t-1} P_{t-1} F_{t-1}^T + Q, \tag{11}$$

where $\hat{x}_{t-1}$ and $\hat{x}_t^-$ are the posterior and prior estimates of the state, and P_{t-1} and P_t^- represent the posterior and prior estimation error covariance matrices.

Correction:

In the correction step, the Kalman gain is computed as:

$$K_t = P_t^- H_t^T (H_t P_t^- H_t^T + R)^{-1}. \tag{12}$$

Then, the posterior state estimate, $\hat{x}_t$, is obtained as:

$$\hat{x}_t = \hat{x}_t^- + K_t(y_t - \hat{y}_t), \tag{13}$$

where y_t represents the actual measurement, and $\hat{y}_t$ denotes the model predicted output value, which can be calculated through:

$$\hat{y}_t = H_t \hat{x}_t^-, \tag{14}$$

Then, the posterior estimation error covariance is

$$P_t = (I - K_t H_t) P_t^-. \tag{15}$$

B. The Kalman-GSINDy Algorithm

When integrating the Kalman filter and the GSINDy algorithm, the model parameters, Ξ, are considered as states to estimate using the Kalman filter. Inspired by library construction step in the SINDy, in the Kalman-GSINDy, the inputs to the Kalman filter are the values of library terms in $\Theta(\mathbf{X})$ at each time instant.

Similar to the Kalman filter, the Kalman-GSINDy can be divided into prediction and correction steps. The prediction in the Kalman-GSINDy is

$$\hat{\Xi}_t^- = F_t \hat{\Xi}_{t-1}, \tag{16}$$

where $\hat{\Xi}_{t-1}$ and $\hat{\Xi}_t^-$ are the posterior and prior estimates of the parameters corresponding to library model terms, respectively. In the Kalman-GSINDy, the states, which represent digital twin model parameters, are not updated in the prediction step, and hence $F_t = I$. Then, the estimation error covariance is predicted as:

$$P_t^- = F_{t-1} P_{t-1} F_{t-1}^T + Q, \tag{17}$$

Even though the state model, F_t, is time-invariant in the Kalman-GSINDy, the output model, H_t, is time-variant, and it equals the library term values at each time instant. Therefore, at time instant t, H_t equals t^{th} row of $\Theta(\mathbf{X})$ and can be denoted as $\Theta(\mathbf{x}_t)$. Consequently, according to (14), at each time instant, the output prediction is

$$\hat{\mathbf{y}}_t = \Theta(\mathbf{x}_t)\hat{\Xi}_t. \tag{18}$$

In the correction steps of the Kalman-GSINDy, the Kalman gain matrix is calculated as:

$$K_t = P_t^- \Theta(\mathbf{x}_t)^T (\Theta(\mathbf{x}_t) P_t^- \Theta(\mathbf{x}_t)^T + R)^{-1}. \tag{19}$$

Different from the SLS algorithm, which iteratively resolves the regression problem, during each iteration, the Kalman-GSINDy keeps modifying the model parameters using the

innovation sequence. In the Kalman filter, the innovation sequence represents the difference between the real output measurements and the model predictions. After each modification, the states (digital twin model parameters) whose absolute values are smaller than λ are forced to be equal to zero, and the nonzero terms are used to provide predictions so as to promote sparsity. As a consequence, the model parameter estimates are modified q times during the correction,

$$\begin{aligned} &\text{for } k = 1 : q, \\ &\quad \left| \hat{\Xi}_t \right| < \lambda = 0, \\ &\quad \hat{\mathbf{y}}_t = \Theta(\mathbf{x}_t)\hat{\Xi}_t \ , \\ &\quad \hat{\Xi}_t = \hat{\Xi}_t \ + K_t(\mathbf{y}_t \quad \hat{\mathbf{y}}_t), \\ &\quad \hat{\Xi}_t \ = \hat{\Xi}_t, \end{aligned} \tag{20}$$

where q is the number of iterations that will enable the estimated model parameters to converge. Usually, the model parameters will converge after 10 iterations. Consequently, q value between 10 and 15 are generally appropriate choices. The λ is magnitude threshold of the model parameters. Selection of λ value will affect performance of the Kalman-GSINDy algorithm. When λ is too large, Kalman-SINDy might miss the necessary model terms and deteriorate prediction accuracy. Small λ values will generally not decrease the prediction accuracy in the Kalman-GSINDy, as it is a recursive algorithm, but will increase the computational cost. Consequently, the value of λ can be increased gradually during tuning process to promote sparsity while preserving prediction accuracy. After each iteration, the new output prediction, $\hat{\mathbf{y}}_t$, is calculated through (18). Afterward, the $\hat{\Xi}_t$ is corrected using the Kalman gain and the innovation sequence, resulting in posterior model parameter estimates of each iteration, $\hat{\Xi}_t$. Then, the prior state estimate, $\hat{\Xi}_t$, is replaced with the posterior state estimate, $\hat{\Xi}_t$, to provide the new output prediction. The output prediction from the last run of iteration is used as the prediction at the current time instant.

After the iterative state corrections, the estimation error covariance is corrected,

$$P_t = (I \quad K_t X_t)P_t \ . \tag{21}$$

Fig. 3 shows an overall illustration of the Kalman-GSINDy procedure. The first step in the Kalman-GSINDy is to identify the objective system of the digital twin construction. Then, necessary input, output variables are identified. After identifying the input variables, a model term library can be constructed based on prior knowledge about the process. If no prior knowledge about the objective system is available, usually, a polynomial library is constructed. Since no training data set is required for the Kalman-GSINDy, the initial estimates for model parameters can be set to close to zero for all the parameters. When a new sample is received, the Kalman-GSINDy will start modifying the model parameters and active term selection through an iterative procedure. The final identified active model terms after iterative correction might be different from the initially identified ones. Then, the Kalman-GSINDy prediction is given using the final identified active model terms and the corresponding parameters.

IV. Case Study

In this section, one numerical Lorenz system example as well as one industrial DHT unit case study are analyzed to demonstrate the advantages of applying the Kalman-GSINDy to construct digital twin models.

A. The Lorenz System

The Lorenz system is a three-dimensional ODE describing chaotic dynamics that oscillate alternately around the two weakly unstable fixed points [21], [22]. The Lorenz system is formulated as [12]:

$$\begin{aligned} \underline{x} &= \sigma(y \quad x), \\ \underline{y} &= x(\rho \quad z) \quad y, \\ \underline{z} &= xy \quad \beta z, \end{aligned} \tag{22}$$

where $\sigma = 10$, $\beta = 8/3$, and $\rho = 28$. The initial condition of the Lorenz system can be randomly selected. In this example, the ODE system is solved through the Matlab function, *ode45*, which utilizes the Runge-Kutta method, with time step, $t = 0.01$, and time span, $T = 100$. After generating time-series data of the states, x, y, z, as well as their corresponding derivatives, using the true Lorenz system equations, random noises, whose magnitudes are scaled to 10% standard deviation of the states are added to both the states and their derivatives.

Afterward, the SINDy is applied to identify the system equations from the whole noisy data set. A second-order polynomial library is constructed. The parameter magnitude threshold, λ, is set to 0.03. The SINDy identified active model terms and their corresponding parameters, with the initial condition of $x_0 = \begin{bmatrix} 8 & 8 & 27 \end{bmatrix}$, are given in Table I. The zero parameter values represent inactive model terms, and the bolded values are the reference parameter values. It can be concluded from Table I that even though the whole data set is available to SINDy with a noise level of 10%, the SINDy-identified model deviated significantly from the true one.

TABLE I
Sample SINDy Estimated Active Model Terms and the Corresponding Parameters with References.

	Objective Outputs					
Library Terms	$\underline{x}$		$\underline{y}$		$\underline{z}$	
1	0.46	**0**	-0.48	**0**	-2.57	**0**
x	-2.51	**−10**	21.88	**28**	-0.06	**0**
y	5.33	**10**	2.50	**−1**	0.06	**0**
z	0	**0**	0	**0**	-2.51	$\mathbf{-\frac{8}{3}}$
x^2	0	**0**	0	**0**	0	**0**
xy	0	**0**	0	**0**	0.91	**1**
xz	-0.20	**0**	-0.83	**−1**	0	**0**
y^2	0	**0**	0	**0**	0.05	**0**
yz	0.11	**0**	-0.09	**0**	0	**0**
z^2	0	**0**	0	**0**	0	**0**

In this example, since the Lorenz system is an ODE system, Kalman-GSINDy is equivalent to Kalman-SINDy.

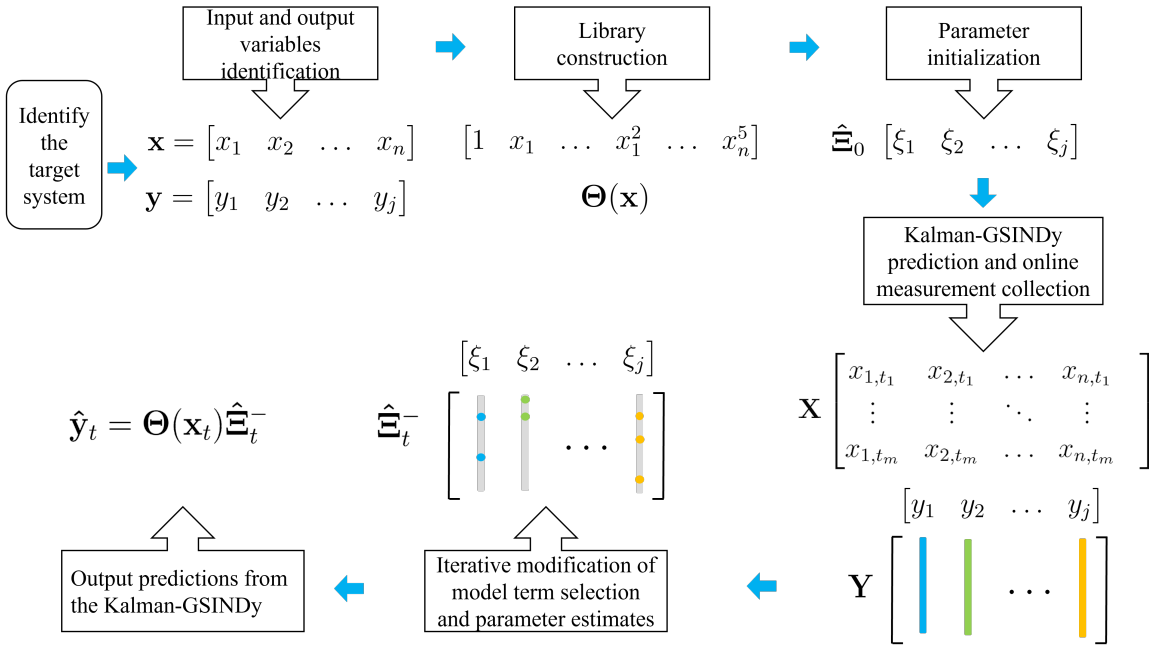

Fig. 3. Graphical illustration about the Kalman-GSINDy steps.

The Kalman-SINDy is applied using the same λ value and the initial parameter estimates are set close to zero with Q and R properly chosen. It is notable that the Kalman-SINDy does not require a data set in advance. It recursively adjusts the selection of active digital twin model terms and the parameter estimates according to the process dynamics.

Monte Carlo performance evaluation is performed using 10,000 simulation runs with data over 10,000 sampling intervals to compare performance between the SINDy and the Kalman-SINDy. For each Monte Carlo simulation run, the added white noise is randomly generated and then scaled. In addition, the initial conditions of the Lorenz system are three randomly generated integers between $\begin{bmatrix} 10, & 10 \end{bmatrix}$ for each simulation run. The average squared estimation error at each time step over the 10,000 simulation runs from both approaches are plotted in Fig. 4. During the transient time period, randomly initialized model parameters lead to larger estimation errors for the Kalman-SINDy. As the active term selection and the estimated parameters get adjusted over time, the Kalman-SINDy effectively provides objective output predictions with lower estimation error than the SINDy for both $\underline{x}$ and $\underline{y}$ proving high estimation accuracy and robustness to data noise. Both the SINDy and the Kalman-SINDy have low estimation error for $\underline{z}$, however, the SINDy requires significantly more data than the Kalman-SINDy.

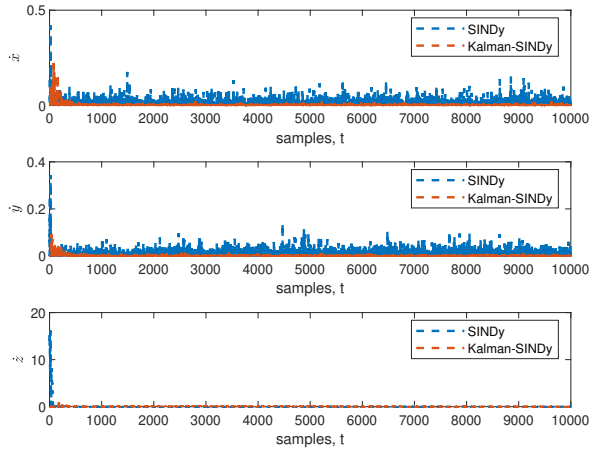

Fig. 4. Average squared estimation error at each time instant over 10,000 Monte Carlo simulation runs from the SINDy and the Kalman-SINDy for the Lorenz system.

B. MIMO Digital Twin Model Construction for a DHT Unit

In the petroleum industry, the DHT unit is used to remove impurities, such as sulfur and nitrogen, from diesel feed so as to satisfy the increasing market and environmental demand for clean fuels. After going through the DHT unit, the heavy oil feed streams are cracked in the presence of catalysts and hydrogen, yielding distillate output and the corresponding by-products. Our objective is to construct a MIMO digital twin model to predict the distillate and by-product yields from related process variables. Overall, 21 process variables are identified as inputs, such as input stream concentration, input flowrates, etc., and 6 output variables are desired to be predicted. Consequently, the GSINDy instead of SINDy is required to construct the MIMO digital twin model.

In this case study, a combined first-order polynomial and trigonometric model library is constructed for model term selection,

$$\boldsymbol{\Theta}(\mathbf{X}) = \begin{bmatrix} 1 & \mathbf{X} & \cos(\mathbf{X}) \end{bmatrix}, \tag{23}$$

where $\mathbf{X} = \begin{bmatrix}\mathbf{x}_{t_1} & \mathbf{x}_{t_2} & \dots & \mathbf{x}_{t_m}\end{bmatrix}^T$, and $\mathbf{x} = \begin{bmatrix}x_1 & x_2 & \dots & x_{21}\end{bmatrix}$. Daily data over a nine-year period are used for analysis, and all the data are normalized for proprietary reasons. Since no noise will be manually added in this example, the prediction performance of the GSINDy is tested on the test data set, which is made up of 30% of the total samples, and the rest of the data are used for training. The Kalman-GSINDy does not require a training data set, and hence it is directly applied to the same test data set, with Q and R appropriately chosen, $\lambda = 0.08$, and the initial parameters set close to zero.

Table II summarizes the test performance for the GSINDy and Kalman-GSINDy using the same λ value in terms of the mean squared error (MSE). It can be concluded from Table II that the Kalman-GSINDy is able to provide more accurate output predictions than the GSINDy identified, time-invariant MIMO model. In addition, the Kalman-GSINDy does not require a training data set as the GSINDy does. As a result, even under small data sets, the Kalman-GSINDy will provide better performance.

TABLE II
MSE FROM THE GSINDY AND KALMAN-GSINDY FOR DHT UNIT OBJECTIVE OUTPUTS PREDICTION.

Objective Outputs (V_{in}=V_{out};%)	**GSINDy**	**Kalman-GSINDy**
Diesel Yield	0.1190	0.0191
Gasoline Yield	0.2110	0.0200
Butane Yield	0.1565	0.0244
Propane Yield	0.0822	0.0276
Ethane Yield	0.1385	0.0287
Methane Yield	0.1725	0.0195

V. CONCLUSIONS

In this paper, a time-variant Kalman-GSINDy digital twin modeling approach is proposed to automatically construct time-variant digital twin models from process data. The GSINDy is a generalization of SINDy, making it applicable to general MIMO problems. The Kalman-GSINDy takes advantage of both the GSINDy and the Kalman filter for online adaptation of the model terms and the respective parameters so as to improve the prediction accuracy. In addition to the prediction accuracy improvement, compared to the GSINDy, the Kalman-GSINDy does not require a large amount of process data, as it adjusts its estimates over time as samples accumulate. However, in general, adequate samples are required to build acceptable SINDy or GSINDy models, and the model accuracy is sensitive to noise in the data. The numerical Lorenz system example and the DHT unit industrial case study demonstrate the advantages of the Kalman-GSINDy. The proposed algorithm can be applied to a wide range of industrial processes to automatically identify their digital twin models and adapt them with time.

REFERENCES

[1] W. Fleming, "Overview of automotive sensors," vol. 1, no. 4, pp. 296–308, 2001, doi: 10.1109/7361.983469.

[2] J. R. Stetter, W. R. Penrose, and S. Yao, "Sensors, chemical sensors, electrochemical sensors, and ECS," vol. 150, no. 2, p. S11, 2003, doi: 10.1149/1.1539051.

[3] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019, doi: 10.1109/tii.2018.2873186.

[4] C. Cimino, E. Negri, and L. Fumagalli, "Review of digital twin applications in manufacturing," vol. 113, p. 103130, Dec. 2019, doi: 10.1016/j.compind.2019.103130.

[5] M. Day, "Discussing digital twins," [Online]. Available: https://aecmag.com/features/discussing-digital-twins/, 2020.

[6] B. Schleich, N. Anwer, L. Mathieu, and S. Wartzack, "Shaping the digital twin for design and production engineering," *CIRP Annals*, vol. 66, no. 1, pp. 141–144, 2017, doi: 10.1016/j.cirp.2017.04.040.

[7] A. Pal, W. Wei, L. Zhu, Y. Wang, and G. Zhu, "Dynamic system identification for a nonlinear vehicle model using *q*-markov cover under different operational conditions," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 2830–2835, doi: 10.23919/acc50511.2021.9482623.

[8] G. N. Schroeder, C. Steinmetz, C. E. Pereira, and D. B. Espindola, "Digital twin data modeling with AutomationML and a communication methodology for data exchange," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 12–17, 2016, doi: 10.1016/j.ifacol.2016.11.115.

[9] F. Tao, M. Zhang, Y. Liu, and A. Nee, "Digital twin driven prognostics and health management for complex equipment," *CIRP Annals*, vol. 67, no. 1, pp. 169–172, 2018, doi: 10.1016/j.cirp.2018.04.055.

[10] W. Xiang, "Data-driven modeling of switched dynamical systems via extreme learning machine," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 852–857, doi: 10.23919/acc50511.2021.9483234.

[11] M. Pratama, M. J. Er, X. Li, R. J. Oentaryo, E. Lughofer, and I. Arifin, "Data driven modeling based on dynamic parsimonious fuzzy neural network," vol. 110, pp. 18–28, June 2013, doi: 10.1016/j.neucom.2012.11.013.

[12] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016, doi: 10.1073/pnas.1517384113.

[13] E. J. LoCicero and L. Bridgeman, "Sparsity promoting H_2-conic control," *IEEE Control Systems Letters*, vol. 5, no. 4, pp. 1453–1458, 2021, doi: 10.1109/lcsys.2020.3039712.

[14] S. H. Rudy, J. N. Kutz, and S. L. Brunton, "Deep learning of dynamics and signal-noise decomposition with time-stepping constraints," *Journal of Computational Physics*, vol. 396, pp. 483–506, Nov. 2019, doi: 10.1016/j.jcp.2019.06.056.

[15] K. Kaheman, S. L. Brunton, and J. N. Kutz, "Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data," *arXiv preprint arXiv:2009.08810*, 2020.

[16] S. Li, E. Kaiser, S. Laima, H. Li, S. L. Brunton, and J. N. Kutz, "Discovering time-varying aerodynamics of a prototype bridge by sparse identification of nonlinear dynamical systems," *Physical Review E*, vol. 100, no. 2, 2019, doi: 10.1103/physreve.100.022220.

[17] A. Longhini, M. Perbellini, S. Gottardi, S. Yi, H. Liu, and M. Zorzi, "Learning the tuned liquid damper dynamics by means of a robust EKF," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 60–65, doi: 10.23919/acc50511.2021.9483172.

[18] Z. Huang, D. Zhang, L. D. Couto, Q.-H. Yang, and S. J. Moura, "State estimation for a zero-dimensional electrochemical model of lithium-sulfur batteries," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 3114–3119, doi: 10.23919/acc50511.2021.9483225.

[19] K. Champion, P. Zheng, A. Y. Aravkin, S. L. Brunton, and J. N. Kutz, "A unified sparse optimization framework to learn parsimonious physics-informed models from data," *IEEE Access*, vol. 8, pp. 169 259–169 271, 2020, doi: 10.1109/access.2020.3023625.

[20] D. Simon, *Optimal state estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sonss, 2006, doi: 10.1002/0470045345.

[21] C. Sparrow, *The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors*. Springer New York, 1982, doi: 10.1007/978-1-4612-5767-7.

[22] X. Wang and M. Wang, "A hyperchaos generated from lorenz system," *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 14, pp. 3751–3758, 2008, doi: 10.1016/j.physa.2008.02.020.