Simultaneous Digital Twin Identification and Signal-Noise Decomposition through Modified Generalized Sparse Identification of Nonlinear Dynamics

Jingyi Wang[a], Jesús Moreira[b], Yankai Cao[a] and R. Bhushan Gopaluni[a,*]

[a]Department of Chemical and Biological Engineering, University of British Columbia ,Vancouver, BC V6T 1Z3, Canada
[b]Digital, Innovation and Lean Capability Development Team, Imperial Oil Company, Calgary, AB T2C 5N1, Canada

ARTICLE INFO

Keywords:
Digital twin
Measurement noise
Modelling
Sparse identification of nonlinear dynamics
System identification

ABSTRACT

A digital twin provides a digital replication of a physical system for remote monitoring, viewing, and control objectives. It has the potential to reshape the future of industrial processes, hence paving the way for smart manufacturing. Automatic system identification techniques that are robust to measurement noise are critical for the development of high-fidelity digital twins and their applications. By establishing a sparse regression framework, the sparse identification of nonlinear dynamics (SINDy) algorithm automatically determines the parsimonious governing equations for physical systems. However, there are some major challenges associated with using SINDy to identify digital twin models. First, the SINDy is restricted to solving the ordinary differential equation (ODE) and partial differential equation (PDE) problems. Second, measurement noise may significantly deteriorate the performance of SINDy. In this paper, the generalized SINDy (GSINDy) algorithm is first introduced to enlarge the SINDy's applicable range. Then, the modified GSINDy (MGSINDy) algorithm is proposed, in which an objective function is constructed to simultaneously identify the digital twin input time-series dynamics model and output model while separating noise from the noisy input. Two numerical examples and one industrial case study are analyzed to demonstrate the advantages of applying the proposed MGSINDy to construct digital twin models. Furthermore, the proposed algorithm can be integrated with the existing SINDy-based online model-adjusting frameworks to become online-adjustable.

1. Introduction

The recent advances in computing technology allow for faster data processing, larger computational power, and the use of sophisticated algorithms and thus enabling the digital representation of a physical system for the purposes of prediction, monitoring, and control (S Wang et al., 2016). Digital twins provide users with computer-based digital representations of physical systems that imitate system behaviours and allow digital interactions with real-world processes. A digital twin receives process data, asset information, record tags, and sensor conditions from the real system. As a result, the digital twin integrates information from disparate system sections and processes real-time data from various sensors or devices to provide estimates of objective outputs and insights on operational conditions (Kender et al., 2021; T Wang et al., 2022; Aversano et al., 2019). By connecting the physical and virtual spaces, the digital twin becomes the promising technology to achieve industrial 4.0 (Kaur et al., 2019; Tao et al., 2018; Cavone et al., 2022; Hung et al., 2022).

Digital twins have been broadly applied to enhance manufacturing performance across various industries. It is stated in (Jimenez et al., 2019) that the digital twins in medical settings are advantageous and will elevate the medical cyber-physical system (MCPS). In (T Wang et al., 2022), the digital twin technique was used to provide a framework for the deployment of real and virtual spaces inside an automated conveyor system in the context of a smart factory. In (Papanagnou, 2019), a digital twin was constructed for an assembly line. In this design, the internet of things (IoT)-captured data is fed back to the digital twin to enhance the online manufacturing performance. A digital twin is used to re-engineer an aircraft structural life forecast by integrating individual physical models into a complete picture of the aircraft (Tuegel et al., 2011).

Digital twin modelling is the vital step in enabling a digital twin to accurately simulate the real physical behaviour in order to understand the system conditions and anticipate output precisely (Yin, Bo, et al., 2021). According to the analysis in (Wright and Davidson, 2020), comparing with general system identification, an effective digital twin identification should fulfill three criteria. First, an effective digital twin model should incorporate sufficient first-principles knowledge. A completely data-driven model usually has a limited domain of applicability (Bangi and Kwon, 2023). Including first-principles knowledge will improve the model's interpretability and enlarge its domain of applicability. Second, a digital twin model should be sufficiently accurate to be able to reflect the process dynamics precisely. Third, a digital twin model should be online-adjustable in response to the process dynamic changes.

While much work has been expended on introducing the digital twin concept and defining its architecture, automatic digital twin modelling from raw sensory data is not yet a widespread practice. An effective, robust, automatic digital twin identification technique will reduce modelling time

*Corresponding author.
E-mail address: bhushan.gopaluni@ubc.ca (R. B. Gopaluni).

consumption and minimize the time delay between data collecting and digital twin creation for a manufacturing process (Uhlemann et al., 2017).

Data-driven approaches for system dynamics discovery are pushing the boundaries of system identification and providing an extraordinary opportunity for automatic digital twin modelling (Kaiser et al., 2018). However, typical data-driven identification techniques have several common issues, such as overfitting, lack of interpretability, and sensitivity to data quality (Yin and Huang, 2022). Real-time measurements are contaminated by noise, which will deteriorate system identification accuracy. This paper proposes an automatic, noise-robust digital twin identification approach that promotes sparsity to strike a compromise between model complexity and accuracy while improving model interpretability.

The sparse identification of nonlinear dynamics (SINDy) proposed in (Brunton et al., 2016) allows for the automated discovery of dynamic systems' governing equations from abundant data. The SINDy consists of three major phases. Collecting time-series data is the initial stage. Then, the second stage is to build a model term library. At this stage, if potential first-principles model terms are available, they can be directly included in the library. The flexible library construction procedure allows SINDy to combine first-principles knowledge and data-driven techniques (Raviprakash et al., 2022) to increase the model's interpretability and domain of applicability. Since the number of model terms inside the library is usually numerous, the last phase of SINDy is to solve a sparse regression problem between the library model terms and objective outputs to select active model terms from the library. By solving the sparse regression problem, parsimonious models will be generated to minimize overfitting, hence improving the interpretability of the discovered models (Kaiser et al., 2018).

SINDy-based algorithms have been applied extensively in identifying chemical process models, including continuous stirred tank reactor (CSTR) (Bhadriraju, Narasingam, et al., 2019), distillation column (Subramanian et al., 2021), hydraulic fracturing (Narasingam and Kwon, 2018), and isothermal batch reactor (Abdullah et al., 2021). In addition to chemical processes, SINDy-based algorithms have also been applied to fluid dynamics (Brunton et al., 2016), physics (K P Champion et al., 2019), and biology (Mangan et al., 2016) to identify the governing equations.

Although SINDy and its variants have been widely utilized to discover governing equations in a variety of fields, there are still several challenges of using SINDy to construct general digital twin models. First, the SINDy is restricted to solving the ordinary differential equation (ODE) and partial differential equation (PDE) problems, in which the output is the derivative or partial derivative of the input and the number of input and output variables is equal. In (J Wang et al., 2022), the authors earlier extended the SINDy to the generalized SINDy (GSINDy), making it applicable to general multi-input multi-output (MIMO) digital twin identification problems.

Second, the real industrial measurements are contaminated by noise, which deteriorates the accuracy of SINDy-identified digital twin models, resulting in deviations between digital twin simulations and actual system conditions (Shardt and Huang, 2013; Xu et al., 2008). The ensemble-SINDy (E-SINDy) technique was proposed in (Fasel et al., 2021), conducting sparse regression on bootstraps and then finding an ensemble of SINDy models to make the algorithm robust to noisy and limited data. In (Kaheman et al., 2020), the modified SINDy (MSINDy) algorithm was proposed to simultaneously denoise process data and discover nonlinear dynamics of the state. In this paper, the modified GSINDy (MGSINDy) is developed with the utilization of both SINDy and GSINDy. The proposed algorithm simultaneously detects the digital twin input time-series dynamics model and output model while denoising the noisy input. By utilizing the proposed approach, the robustness and accuracy of digital twin identification will be enhanced.

In addition to time-invariant system identification, several SINDy-based online model-adjusting frameworks have been developed. In (Bhadriraju, Bangi, et al., 2020), an operable adaptive sparse identification of systems (OASIS) was proposed to first identify multiple SINDy-based sparse models according to different operating conditions, and then train a deep neural network to predict and update the sparse model that should be used for the model-based controller. The OASIS algorithm was subsequently integrated with a risk assessment approach to provide a fault prognosis for chemical processes (Bhadriraju, Kwon, et al., 2021). In (Bhadriraju, Narasingam, et al., 2019), a three-step framework was developed to first apply SINDy offline to identify the model terms and then use ordinary least-squares regression and step-wise feature selection to update model parameters and features online in response to the process dynamic changes. A SINDy-based rapid model recovery framework was proposed in (Quade et al., 2018) to detect model divergence and update the SINDy-identified model to ensure online estimation accuracy. The proposed MGSINDy algorithm can be seamlessly integrated with the above mentioned adaptive implementation frameworks by substituting the SINDy step with the MGSINDy to adjust the MGSINDy-identified models online.

The main contribution of this paper is to propose a novel digital twin identification approach for automatic digital twin construction in the presence of input measurement noise. In this approach, the input time-series dynamics and the output model mapping the denoised input to actual output are simultaneously identified. This simultaneous system identification and measurement denoising procedure is accomplished by solving an optimization problem with an appropriately designed objective function, considering the input measurement noise, the input time-series dynamics, and the sparse output model parameters as optimization variables. The proposed algorithm has the potential to be adjusted online by integrating with the current SINDy-based adaptive system identification frameworks.

The remainder of this paper is organized in the following manner. Section 2 explains the current challenges of applying SINDy to identify general digital twin models, and Section 3 provides an overview of the SINDy and GSINDy algorithms. Section 4 elaborates on the proposed MGSINDy approach. In Section 5, two numerical examples and one case study of an industrial diesel hydrotreating (DHT) unit are investigated to demonstrate the advantages of the proposed MGSINDy approach. Finally, section 6 concludes this paper.

2. Problem Statement

This work focuses on the development of digital twin models for MIMO nonlinear physical systems. As a consequence, we consider the continuous-time, nonlinear system of the form,

$$\begin{aligned} \dot{\mathbf{x}}_t &= f_x(\mathbf{x}_t), \\ \mathbf{y}_t &= f_y(\mathbf{x}_t), \end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ is the input variable, and f_x is a Lipschitz continuous vector field describing the input time-series dynamics; $\mathbf{y} \in \mathbb{R}^{n_y}$ is the output variable, and f_y is the nonlinear output model, mapping the input variables to the output variables. In addition, $t \in \mathbb{R}^m$ represents time instants.

In this study, we would like to simultaneously identify the time-series dynamics, f_x, and the output model, f_y. We assume that the inputs are directly measured, and the outputs are either directly measured or collected from lab analysis. When simulating a real physical system with a digital twin, the input measurements are unavoidably contaminated by noise,

$$\mathbf{X_n} = \mathbf{X} + \mathbf{N}, \tag{2}$$

where $\mathbf{X} = \begin{bmatrix} \mathbf{x}_{t_1} & \mathbf{x}_{t_2} & \dots & \mathbf{x}_{t_m} \end{bmatrix}^T \in \mathbb{R}^{m \times n_x}$ is the noise-free input measurement, and $\mathbf{X_n}$ represents the noise-corrupted input measurement, and $\mathbf{N}$ denotes the input measurement noise, which is assumed to be Gaussian noise in this analysis.

When applying SINDy to identify the digital twin models, there exist some major challenges. As previously stated in Section 1, SINDy is restricted to identifying the ODE and PDE relationships, in which the output is set to be equal to the derivative or partial derivative of input, resulting in equal number of input and output variables. As a result, the SINDy is only applicable to solve for f_x. In (J Wang et al., 2022), the GSINDy is introduced to extend the SINDy's output to general system output measurements and eliminate the constraint on the number of output variables. Consequently, GSINDy is applicable to identify f_y.

In addition, a digital twin simulates the real physical system operations. A well-built digital twin model can correctly mirror the state of the real system. The construction of digital twin models is based on the data collected through sensors or other measurement devices (Dang et al., 2022). In real operations, such measurements are invariably affected by noise, and the measurement noise will deteriorate model identification accuracy from SINDy or GSINDy, resulting in estimation deviations from the actual values (Sun et al., 2021). The objective of this study is to propose a noise-robust, automatic digital twin identification algorithm, which simultaneously identifies the digital twin input time-series dynamics model and output model while denoising the input measurement. In this instance, since the output variable is the prediction objective, the output value is considered accurate and used as a benchmark for evaluating the model accuracy.

3. The SINDy and GSINDy Algorithms

3.1. SINDy

Understanding the time-series dynamics of the input is critical for gaining an insight into the system operational conditions. SINDy uses three phases to determine the nonlinear time-series dynamics of the input, f_x. The first step is to acquire or generate data. In addition to collecting sensor readings, clean data may be generated by solving an ODE or a PDE problem in numerical simulations. Although the input derivative is difficult to measure, it may be calculated using numerical techniques, such as the central difference approach. The second stage is to establish a model term library, which will include prospective model terms, from which the system model may be constructed. This library is highly flexible. If prior knowledge about the system is accessible, particular model terms from first-principles information can be included in the library. Data-driven model terms, such as polynomial, trigonometric, and Fourier terms, are conventional to construct a SINDy model term library. Notably, the first-principles and data-driven terms can be combined to create an overall comprehensive model term library, resulting in a hybrid identification procedure.

An integrated model term library with polynomial and first-principles terms is provided as an example,

$$\mathbf{\Theta_x}(\mathbf{X}) = \begin{bmatrix} \mathbf{1} & \mathbf{X} & \mathbf{X}^{P_2} & \dots & \mathrm{UA\Delta T} & \dots \end{bmatrix}, \tag{3}$$

where $\mathbf{\Theta_x}(\mathbf{X})$ denotes the model term library for input dynamics, and $\mathbf{X}^{P_i}$ represents all polynomial terms of order i. For instance,

$$\mathbf{X}^{P_2} = \begin{bmatrix} x_{1,t_1}^2 & x_{1,t_1}x_{2,t_1} & \dots & x_{2,t_1}^2 & \dots & x_{n_x,t_1}^2 \\ x_{1,t_2}^2 & x_{1,t_2}x_{2,t_2} & \dots & x_{2,t_2}^2 & \dots & x_{n_x,t_2}^2 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{1,t_m}^2 & x_{1,t_m}x_{2,t_m} & \dots & x_{2,t_m}^2 & \dots & x_{n_x,t_m}^2 \end{bmatrix}. \tag{4}$$

In equation (3), UAΔT is a hypothetical first-principles term from heat transfer, where U is the overall heat transfer coefficient, A denotes the heat transfer area, and ΔT represents the temperature difference (Bergman et al., 2011).

SINDy's last phase, after constructing the model term library, is to solve a sparse regression problem to extract

relevant model terms from the library. The regression is formulated as,

$$\dot{\mathbf{X}} = \boldsymbol{\Theta}_{\mathbf{x}}(\mathbf{X})\boldsymbol{\Xi}_{\mathbf{x}}, \tag{5}$$

where $\dot{\mathbf{X}}$ denotes the numerically calculable input derivative, and $\boldsymbol{\Xi}_{\mathbf{x}}$ is the sparse parameter matrix for input dynamics, which provides the users with sparse model term selection and the corresponding parameters for the input time-series dynamics. Typically, the sequential least squares (SLS) regression algorithm is used to solve the sparse regression problem in SINDy, since it forces zero magnitude parameters to promote model sparsity (Brunton et al., 2016; K Champion et al., 2020).

3.2. GSINDy

To enlarge the SINDy's applicable range, in (J Wang et al., 2022), the GSINDy is introduced to relax the restrictions on the SINDy's output. Along with collecting input measurements, GSINDy also collects output measurements over time,

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_{t_1} & \mathbf{y}_{t_2} & \cdots & \mathbf{y}_{t_m} \end{bmatrix}^T, \tag{6}$$

where $\mathbf{Y} \in \mathbb{R}^{m \times n_y}$. Subsequently, the output model term library, $\boldsymbol{\Theta}_{\mathbf{y}}(\mathbf{X})$ is constructed. Then, the following sparse regression problem is to be solved to achieve the sparse parameter matrix for output model, $\boldsymbol{\Xi}_{\mathbf{y}}$,

$$\mathbf{Y} = \boldsymbol{\Theta}_{\mathbf{y}}(\mathbf{X})\boldsymbol{\Xi}_{\mathbf{y}}. \tag{7}$$

Fig. 1 illustrates the GSINDy method graphically. Both SINDy and GSINDy use the SLS to solve the sparse regression problem between objective outputs and library model terms through an iterative procedure. In each iteration, the SLS forces the parameters in $\boldsymbol{\Xi}$ whose magnitudes are smaller than the threshold, λ, to be zero. The nonzero parameters indicate that their corresponding library model terms are active. Subsequently, another iteration is performed between the active model terms and the objective outputs until convergence. Usually, ten iterations are sufficient for parameter convergence. By solving the sparse regression problem, the majority of the entries in the sparse parameter matrix, $\boldsymbol{\Xi}$, will equal to zero, providing us with a sparse governing equation. In Fig. 1, the colourful dots in the $\boldsymbol{\Xi}_{\mathbf{y}}$ matrix reflect the identified active model terms for the individual output variable.

4. The MGSINDy Algorithm

Section 3, introduced the SINDy and GSINDy approaches. In this section, the MGSINDy is proposed using SINDy to identify the time-series dynamics of input variables, f_x, while using GSINDy to estimate the relationship between denoised input and actual output, f_y. In the MGSINDy, a time-stepping constraint is used to estimate the input noise and ensure the consistency of estimated and true input time series dynamics (Kaheman et al., 2020; Rudy et al., 2019). The simultaneous digital twin identification and

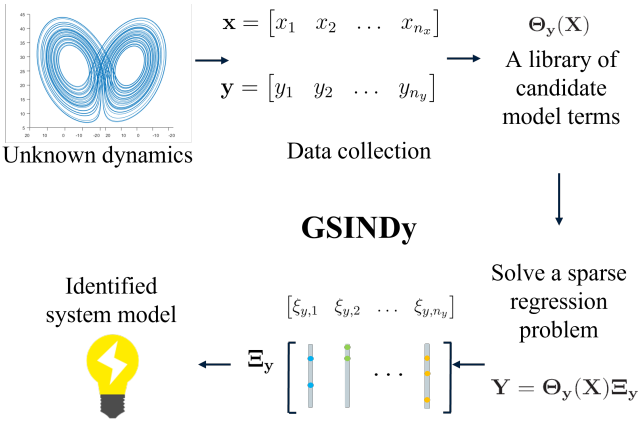

Fig. 1: Graphical illustration of the GSINDy procedure.

signal-noise decomposition is achieved through an appropriately designed objective function. The three sections of the designed objective function in MGSINDy are introduced in detail in the following subsections.

4.1. Input Derivative Estimation Error

In MGSINDy, the input time-series dynamics are estimated by

$$f_x(\mathbf{X}) = \boldsymbol{\Theta}_{\mathbf{x}}(\mathbf{X})\boldsymbol{\Xi}_{\mathbf{x}}. \tag{8}$$

When performing digital twin identification on a genuine physical system, the input measurements are unavoidably contaminated by noise. Then, the SINDy's estimation for the nonlinear time-series dynamics of a noisy input can be written in the form of

$$\dot{\mathbf{X}}_{\mathbf{n}} = \boldsymbol{\Theta}_{\mathbf{x}}(\mathbf{X}_{\mathbf{n}})\boldsymbol{\Xi}_{\mathbf{x}} = \boldsymbol{\Theta}_{\mathbf{x}}(\mathbf{X} + \mathbf{N})\boldsymbol{\Xi}_{\mathbf{x}}, \tag{9}$$

where $\dot{\mathbf{X}}_{\mathbf{n}}$ is the derivative of noisy input, $\boldsymbol{\Theta}_{\mathbf{x}}(\mathbf{X}_{\mathbf{n}})$ is the model term library constructed based on the noisy input.

This allows for the formulation of the derivative estimation error of input, $e_{x,d}$, minimizing the difference between the SINDy-estimated input derivative and the numerically computed noise-free input derivative,

$$\begin{aligned} e_{x,d} &= \left\| \dot{\hat{\mathbf{X}}} - \boldsymbol{\Theta}_{\mathbf{x}}(\hat{\mathbf{X}})\boldsymbol{\Xi}_{\mathbf{x}} \right\|_2^2 \\ &= \left\| \dot{\hat{\mathbf{X}}} - \boldsymbol{\Theta}_{\mathbf{x}}(\mathbf{X}_{\mathbf{n}} - \hat{\mathbf{N}})\boldsymbol{\Xi}_{\mathbf{x}} \right\|_2^2. \end{aligned} \tag{10}$$

In equation (10), $\hat{\mathbf{N}}$ and $\boldsymbol{\Xi}_{\mathbf{x}}$ are two uncorrelated optimization parameters, which will result in trivial solutions (Rudy et al., 2019) using noise to capture all the input dynamics. In order to solve this optimization problem to refine estimates for both the input measurement noise, $\hat{\mathbf{N}}$, and the SINDy model parameter, $\boldsymbol{\Xi}_{\mathbf{x}}$, time-stepping constraints are applied to form the input simulation error, as described in (Rudy et al., 2019), and (Kaheman et al., 2020).

4.2. Input Simulation Error

Equation (8) uses SINDy to estimate the input vector field, f_x. The discrete-time map from $\mathbf{x}_j$ to $\mathbf{x}_{j+1}$ may be found by integrating this vector field over one step of time, t_j to t_{j+1},

$$\mathbf{x}_{j+1} = \mathbf{F}_{\mathbf{x}}(\mathbf{x}_j) = \mathbf{x}_j + \int_{t_j}^{t_{j+1}} \mathbf{\Theta}_{\mathbf{x}}(\mathbf{x}(\tau))\mathbf{\Xi}_{\mathbf{x}}\, d\tau, \tag{11}$$

where $\mathbf{F}_{\mathbf{x}}$ denotes the one time step integration of input vector field, f_x. This integration can be performed for q time steps to integrate the current input forward or backward. For example,

$$\mathbf{x}_{j+q} = \mathbf{F}_{\mathbf{x}}{}^{q}(\mathbf{x}_j) = \mathbf{x}_j + \int_{t_j}^{t_{j+q}} \mathbf{\Theta}_{\mathbf{x}}(\mathbf{x}(\tau))\mathbf{\Xi}_{\mathbf{x}}\, d\tau. \tag{12}$$

With the availability of the SINDy-estimated, nonlinear input time-series dynamics, $\hat{f}_x$, we can also use the numerical method to determine the denoised input value in the future and previous q steps, $\mathbf{x}_{j+q}$ and $\mathbf{x}_{j-q}$. In this paper, we use the 4$^{\text{th}}$-order Runge-Kutta (RK4) approach to compute the denoised estimate of $\mathbf{x}_{j+q}$,

$$\begin{aligned}\hat{\mathbf{x}}_{j+q} &= \hat{\mathbf{F}}_{\mathbf{x}}^{q}(\hat{\mathbf{x}}_j) \\ &= \hat{\mathbf{x}}_j + \int_{t_j}^{t_{j+q}} \mathbf{\Theta}_{\mathbf{x}}(\hat{\mathbf{x}}(\tau))\mathbf{\Xi}_{\mathbf{x}}\, d\tau.\end{aligned} \tag{13}$$

In this instance, we are able to generate two estimates of the denoised input at time instant $j+q$. One is the result of subtracting the noise estimate from the noisy input measurement, $\hat{\mathbf{x}}_{j+q} = \mathbf{x}_{\mathbf{n}j+q} - \hat{\mathbf{n}}_{j+q}$, and the other is the result of the q-step simulation from $\hat{\mathbf{x}}_j$ using the RK4 algorithm, $\hat{\mathbf{x}}_{j+q} = \hat{\mathbf{F}}_{\mathbf{x}}^{q}(\hat{\mathbf{x}}_j)$. This provides us with the time-stepping constraint to optimize the combination between $\hat{\mathbf{N}}$ and $\mathbf{\Xi}_{\mathbf{x}}$.

Equation (14) further elaborates on the integration of input dynamics estimation and noise estimation based on time-stepping constraint,

$$\hat{\mathbf{x}}_{\mathbf{n}j+q} = \overbrace{\hat{\mathbf{F}}_{\mathbf{x}}^{q}(\underbrace{\mathbf{x}_{\mathbf{n}j} - \hat{\mathbf{n}}_j}_{\hat{\mathbf{x}}_j})}^{\hat{\mathbf{x}}_{j+q}} + \hat{\mathbf{n}}_{j+q}. \tag{14}$$

Involving the time-stepping constraint to the overall objective function of the MGSINDy will not only provide us with additional constraint to prevent trivial solutions but also enforce the consistency of the estimated and true input time-series dynamics.

Consequently, at time instant j, the error associated with the input estimate from past q time steps to the future q time steps is

$$e_{x,s,j} = \sum_{i=-q, i\neq 0}^{q} w_i \left\| \mathbf{x}_{\mathbf{n}j+i} - \hat{\mathbf{n}}_{j+i} - \hat{\mathbf{F}}_{\mathbf{x}}^{i}(\hat{\mathbf{x}}_j) \right\|_2^2, \tag{15}$$

where $e_{x,s}$ is the input simulation error, and $w_i = 0.9^{|i|-1}$ is a weighting parameter that exponentially decreases the importance of the input simulation error at each time instant away from i. This weighting parameter enables us to place a higher premium on recent estimates and a lower premium on distant forecasts. The overall input simulation error over the entire estimation trajectory is

$$\begin{aligned}e_{x,s} &= \sum_{j=q+1}^{m-q} e_{x,s,j} \\ &= \sum_{j=q+1}^{m-q} \sum_{i=-q, i\neq 0}^{q} w_i \left\| \mathbf{x}_{\mathbf{n}j+i} - \hat{\mathbf{n}}_{j+i} - \hat{\mathbf{F}}_{\mathbf{x}}^{i}(\hat{\mathbf{x}}_j) \right\|_2^2.\end{aligned} \tag{16}$$

4.3. Output Prediction Error

To determine the input time-series dynamics and the output model at the same time, an additional error is included in the overall objective function. Similar to equation (10), the output prediction error, e_y, representing the difference between the output measurement and its corresponding GSINDy prediction, is

$$\begin{aligned}e_y &= \left\| \mathbf{Y} - \mathbf{\Theta}_{\mathbf{y}}(\hat{\mathbf{X}})\mathbf{\Xi}_{\mathbf{y}} \right\|_2^2 \\ &= \left\| \mathbf{Y} - \mathbf{\Theta}_{\mathbf{y}}(\mathbf{X}_{\mathbf{n}} - \hat{\mathbf{N}})\mathbf{\Xi}_{\mathbf{y}} \right\|_2^2,\end{aligned} \tag{17}$$

where $\mathbf{Y}$ denotes the measured output value and is regarded as accurate; $\mathbf{\Theta}_{\mathbf{y}}$ is the model term library for output prediction, and $\mathbf{\Xi}_{\mathbf{y}}$ is the corresponding sparse parameter matrix.

Up to now, we have introduced three categories of errors. Among these error equations, there exist three optimization parameters, $\mathbf{\Xi}_{\mathbf{x}}$, $\hat{\mathbf{N}}$, and $\mathbf{\Xi}_{\mathbf{y}}$. By aggregating the three errors, we are able to formulate the overall objective function of the optimization problem in the MGSINDy,

$$\begin{aligned}\mathcal{L}(\mathbf{\Xi}_{\mathbf{x}}, \mathbf{\Xi}_{\mathbf{y}}, \hat{\mathbf{N}}) &= e_{x,d} + e_{x,s} + e_y \\ &= \left\| \dot{\hat{\mathbf{X}}} - \mathbf{\Theta}_{\mathbf{x}}(\hat{\mathbf{X}})\mathbf{\Xi}_{\mathbf{x}} \right\|_2^2 \\ &\quad + \sum_{j=q+1}^{m-q} \sum_{i=-q, i\neq 0}^{q} w_i \left\| \mathbf{x}_{\mathbf{n}j+i} - \hat{\mathbf{n}}_{j+i} - \hat{\mathbf{F}}_{\mathbf{x}}^{i}(\hat{\mathbf{x}}_j) \right\|_2^2 \\ &\quad + \left\| \mathbf{Y} - \mathbf{\Theta}_{\mathbf{y}}(\hat{\mathbf{X}})\mathbf{\Xi}_{\mathbf{y}} \right\|_2^2.\end{aligned} \tag{18}$$

Subsequently, the optimization problem can be summarized as

$$\mathbf{\Xi}_{\mathbf{x}}, \mathbf{\Xi}_{\mathbf{y}}, \hat{\mathbf{N}} = \underset{\mathbf{\Xi}_{\mathbf{x}}, \mathbf{\Xi}_{\mathbf{y}}, \hat{\mathbf{N}}}{\operatorname{argmin}} \mathcal{L}(\mathbf{\Xi}_{\mathbf{x}}, \mathbf{\Xi}_{\mathbf{y}}, \hat{\mathbf{N}}), \tag{19}$$

where the elements within $\mathbf{\Xi}_{\mathbf{x}}$ and $\mathbf{\Xi}_{\mathbf{y}}$ whose magnitudes are less than the thresholding parameters λ_x and λ_y will be forced to equal zero to form a thresholding optimization procedure.

In the MGSINDy, the optimization technique is integrated with the SLS to build the thresholding optimization

procedure (Kaheman et al., 2020). At the first optimization loop, the noise estimates are initialized as zeros or small values close to zero. In the overall thresholding optimization procedure, the number of optimization loops is user-defined. Generally, the estimation error diminishes as the number of optimization loops increases and the algorithm converges. Following each optimization loop, the user will acquire the estimated $\boldsymbol{\Xi}_{\mathbf{x}}$, $\hat{\mathbf{N}}$, and $\boldsymbol{\Xi}_{\mathbf{y}}$. After that, the denoised input, $\hat{\mathbf{X}}$, is calculated and used to build the model term libraries, $\boldsymbol{\Theta}_{\mathbf{x}}(\hat{\mathbf{X}})$ and $\boldsymbol{\Theta}_{\mathbf{y}}(\hat{\mathbf{X}})$, while the $\dot{\hat{\mathbf{X}}}$ is computed numerically. In the meanwhile, the thresholding parameters λ_x and λ_y are used to determine the active model terms from the parameters given in $\boldsymbol{\Xi}_{\mathbf{x}}$ and $\boldsymbol{\Xi}_{\mathbf{y}}$, respectively. Afterward, regressions are performed between the active terms inside $\boldsymbol{\Theta}_{\mathbf{x}}(\hat{\mathbf{X}})$, $\boldsymbol{\Theta}_{\mathbf{y}}(\hat{\mathbf{X}})$ and $\dot{\hat{\mathbf{X}}}$, $\mathbf{Y}$, respectively.

Typically, the model term library for SINDy-based algorithms contains a number of nonlinear terms, creating a non-convex optimization problem for the MGSINDy. When constructing the model term library, $\boldsymbol{\Theta}(\hat{\mathbf{X}})$, the complexity of the library should be progressively enhanced to avoid the SINDy-based algorithms identifying superfluous terms. In the case of generating a large-size model term library, the computational time required to solve the optimization problem would increase. Nevertheless, since there is no specific constraints on the overall objective function, the MGSINDy optimization problem can be solved by using the Adam optimizer with the capability to converge to a local minimum. In the case of converging to a poor local minimum, multi-start strategy can be conducted by re-initializing the noise estimates.

In addition, the sparsity and accuracy of the identified models are sensitive to the value of thresholding parameter, λ. If the λ is too small, additional terms might be determined, then, $\boldsymbol{\Xi}$ and $\hat{\mathbf{N}}$ will be easier to stuck in their local minimums. If λ is too large, the algorithm will miss the necessary model terms, resulting in performance degradation. When using the MGSINDy, one may progressively raise the values of λ_x and λ_y, until the performance begins to degrade significantly, at which point the appropriate thresholding parameter value will be determined. Detailed discussion about the effects of data length, number of optimization loops, and values of thresholding parameters can be found in (Kaheman et al., 2020).

Fig. 2 illustrates the proposed MGSINDy algorithm. As shown in this figure, MGSINDy aims to concurrently identify the digital twin input time-series model and output model while isolating the noise from noisy input measurement. The MGSINDy accomplishes this objective by solving an optimization problem with a three-section objective function. The first section is the input derivative estimation error, $e_{x,d}$, minimizing the difference between the SINDy-estimated derivatives and the numerically-calculated, denoised input derivatives. The input simulation error, $e_{x,s}$, is the second section, expressing the time-stepping constraint. The last component of the overall objective function is the output prediction error, e_y, minimizing the difference between the real output measurements and the GSINDy predictions.

5. Case Studies

5.1. Numerical Examples

5.1.1. Rössler Attractor

The Rössler attractor is also known as Rössler system and was first introduced by Otto Rössler in 1970s as a three-dimensional ODE system describing continuous chaotic dynamics (Maris and Goussis, 2015; Gaspard, 2005). The Rössler attractor is defined by the following ODEs:

$$\begin{aligned} \dot{x}_1 &= -x_2 - x_3, \\ \dot{x}_2 &= x_1 + a x_2, \\ \dot{x}_3 &= b x_1 - c x_3 + x_1 x_3, \end{aligned} \tag{20}$$

where x_1, x_2, x_3 are dynamical variables, and a, b, c are parameters. In this study, we will evaluate the MGSINDy's performance using a set of Rössler attractor parameters that is commonly used: $a = 0.2$, $b = 0.2$, $c = 5.7$ (Kuznetsov et al., 2014). The output model is designed as

$$\begin{aligned} y_1 &= -2x_1^2 + 0.5x_3^3, \\ y_2 &= 2x_1 x_2 + x_3^2, \\ y_3 &= x_1 x_3 - x_1 x_2 x_3. \end{aligned} \tag{21}$$

When constructing a polynomial library, the library order is progressively increased. As a result, the second-order and the third-order model term libraries are built for the input time-series dynamics model and the output model, respectively. Owing to the noise-corrupted nature of the input, we use $\mathbf{X}_{\mathbf{n}}$ inside the library representation. Accordingly,

$$\boldsymbol{\Theta}_{\mathbf{x}}(\mathbf{X}_{\mathbf{n}}) = \begin{bmatrix} \mathbf{X}_{\mathbf{n}} & \mathbf{X}_{\mathbf{n}}{}^{P_2} \end{bmatrix}, \tag{22}$$

$$\boldsymbol{\Theta}_{\mathbf{y}}(\mathbf{X}_{\mathbf{n}}) = \begin{bmatrix} \mathbf{X}_{\mathbf{n}} & \mathbf{X}_{\mathbf{n}}{}^{P_2} & \mathbf{X}_{\mathbf{n}}.^{3} & \mathrm{X}_{\mathrm{n},1}\mathrm{X}_{\mathrm{n},2}\mathrm{X}_{\mathrm{n},3} \end{bmatrix}, \tag{23}$$

here $\mathbf{X}_{\mathbf{n}}.^{3}$ only contains the third-order term of each individual input variable, which is a subset of $\mathbf{X}_{\mathbf{n}}{}^{P_3}$.

Traditionally, it requires one application of SINDy and another application of GSINDy to identify both the input time-series dynamics and the output model. Through applying the proposed MGSINDy algorithm, these two types of models are achieved simultaneously with the signal-noise decomposition of input. In this example, the clean input values are generated by solving the ODE system with a time step of dt $= 0.02$, a time span of T $= 25$, and the initial condition is $\mathbf{x}_0 = \begin{bmatrix} 5 & 2 & 15 \end{bmatrix}$. The accuracy of MGSINDy is then evaluated by adding Gaussian noise to the clean input at levels of 5%, 15%, and 25%. The magnitude of the noise is generated according to the percentage of the standard deviation of the clean data. It takes 400, 550, and 600 samples for the MGSINDy to identify the input time-series dynamics and output prediction models with relatively high accuracy at 5%, 15%, and 25%, respectively. In general,

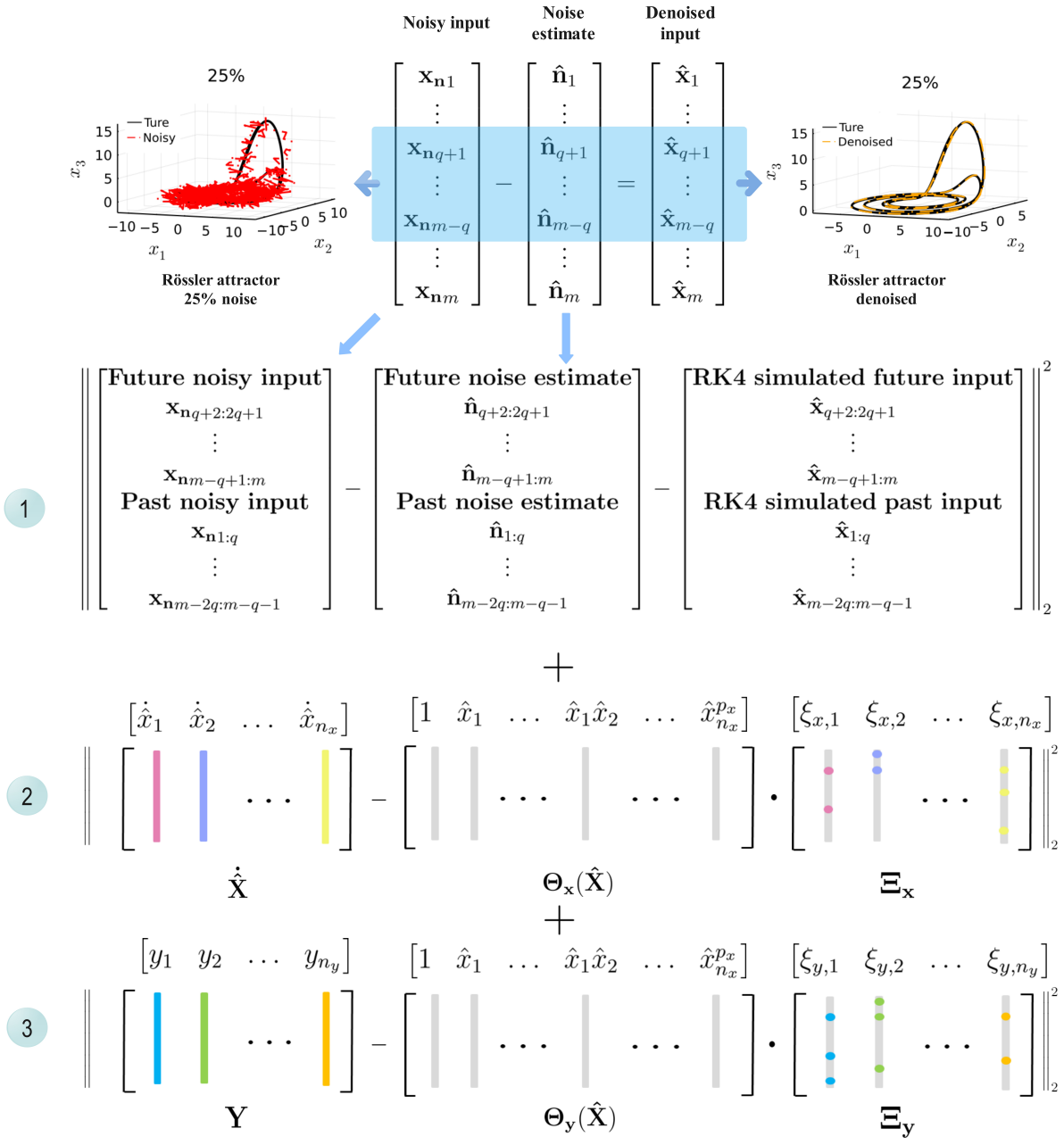

Fig. 2: Graphical illustration of the MGSINDy objective function.

as the noise level increases, more samples will be required to accurately identify the model. The data length requirement and the denoising performance for non-Gaussian noise for the MSINDy-based algorithm are discussed in (Kaheman et al., 2020). In this example, we use T = 25 (1250 samples) for performance demonstration.

Thresholding parameters are set to $\lambda_x = 0.1$ for SINDy and $\lambda_y = 0.4$ for GSINDy. The same parameters are applied in MGSINDy. According to the investigation in (Kaheman et al., 2020), the model identification accuracy will not be notably increased by increasing the prediction step q. However, the computational cost increases linearly with an increase of q. As a consequence, $q = 1$ is an appropriate choice to preserve the accuracy of system identification while minimizing the computational cost. In this example, the number of optimization loops is three. For all examples in this paper, the Adam optimizer with a learning rate of 0.001 is used to solve the optimization problem.

In Fig. 3, the left column shows the clean input data generated by solving the Rössler ODE system and the noise-corrupted data with different noise levels. While the right column shows the denoised input data by applying the MGSINDy algorithm with clean data as references. Comparing the two columns in Fig. 3, we can observe that the MGSINDy successfully denoised the noisy input with varying levels of noise. In Table 1, the SINDy and

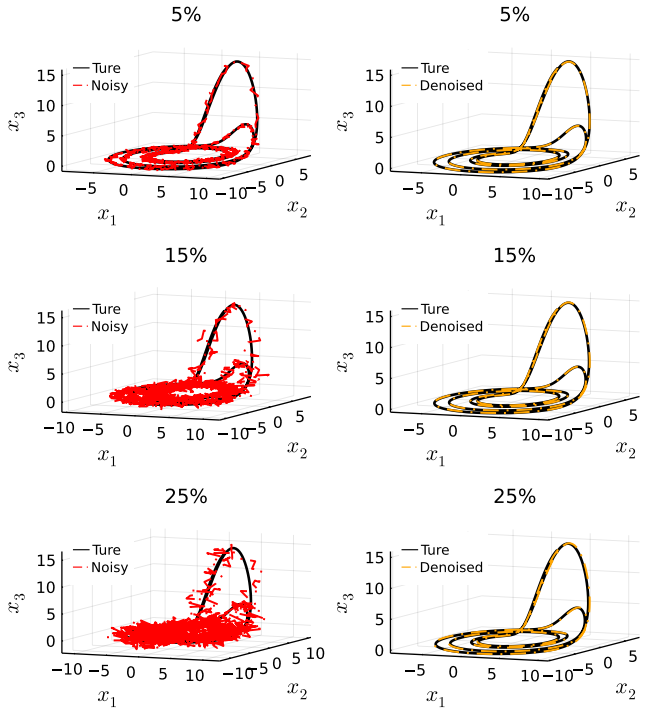

Fig. 3: Graphical illustration of noise-corrupted and MGSINDy-denoised Rössler attractor data with different noise levels using clean data as references.

Table 1
SINDy and MGSINDy-estimated Model Parameters at 25% Noise Level with References.

		Model Library Terms						
	Methods	1	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3
$\dot{x}_1$	Reference	**0**	**0**	**−1**	**−1**	**0**	**0**	**0**
	MGSINDy	0	0	−1.0	−1.0	0	0	0
	SINDy	0	0	−0.9	0	0	0	−0.4
$\dot{x}_2$	Reference	**0**	**1**	**0.2**	**0**	**0**	**0**	**0**
	MGSINDy	0	1	0.2	0	0	0	0
	SINDy	0	1	0.2	0.4	0	0	−0.1
$\dot{x}_3$	Reference	**0.2**	**0**	**0**	**−5.7**	**0**	**1**	**0**
	MGSINDy	0	0	0	−5.1	0	0.9	−0.1
	SINDy	−0.2	0	0	−2.7	0	0.6	−0.3

MGSINDy-estimated model parameters at 25% noise level are presented. Since neither the SINDy nor the MGSINDy identifies the $\mathbf{X}.^2$ terms, these terms are omitted from the table. In the presence of measurement noise, SINDy is more susceptible to mistakenly determining the incorrect model terms than MGSINDy. In addition, the SINDy-estimated parameters show greater deviations from the references than the MGSINDy-estimated values due to noise contamination.

The graphical illustration of output predictions from GSINDy and MGSINDy-identified models are shown in Fig. 4. Without noise-signal decomposition, the GSINDy can only form the model term library using noisy inputs, which will in turn, deteriorate the output model identification accuracy. On the contrary, the MGSINDy estimates input measurement noise, input time-series dynamics, and output model simultaneously. The model term library based on the denoised input enables identifying a more accurate output model and hence improves the output prediction accuracy significantly.

Table 2
ARMSE from the GSINDy and MGSINDy for Rössler System Output Predictions.

Noise level(%)	Variable	GSINDy	MGSINDy
5%	y_1	6.82	7.1e-3
	y_2	3.18	4.1e-3
	y_3	4.07	3.6e-3
15%	y_1	21.34	0.48
	y_2	9.14	0.31
	y_3	10.58	0.15
25%	y_1	34.98	0.83
	y_2	14.89	0.72
	y_3	16.07	0.97

Table 2 presents the quantitative performance comparisons for GSINDy and MGSINDy output predictions. In Table 2, the average root mean squared errors (ARMSE) for 1000 Monte Carlo runs are presented. In each Monte Carlo run, the noise is regenerated. The ARMSE values are calculated as

$$\text{ARMSE} = \sqrt{\frac{1}{L}\frac{1}{K}\sum_{l=1}^{L}\sum_{k=1}^{K}\|e_{k,l}\|^2}, \tag{24}$$

where e represents the error between the reference and the prediction at each time instant for each Monte Carlo run; L represents the number of Monte Carlo simulation runs, which is 1000 in this example, and K is the number of time step within each simulation run, equaling 1250. The ARMSE values comparison presented in Table 2 proves that the MGSINDy makes the system identification procedure more noise-resistant.

5.2. Michaelis-Menten Kinetics

The Michaelis-Menten kinetics is derived from an enzymatic reaction and can be expressed as (Mangan et al., 2016)

$$\dot{x} = 0.6 - \frac{1.5x}{0.3 + x}, \tag{25}$$

which is a nonlinear ODE. Two strategies might be used to detect this nonlinear ODE model using SINDy-based methodologies. To begin, assuming prior knowledge is available, and we know that the objective function is rational, we can therefore create a model term library involving the rational terms, such as

$$\boldsymbol{\Theta}_{\mathbf{x}}(\mathbf{X}) = \left[\mathbf{1} \quad \mathbf{X} \quad \frac{\mathbf{X}}{\mathbf{a}+\mathbf{X}}\right], \tag{26}$$

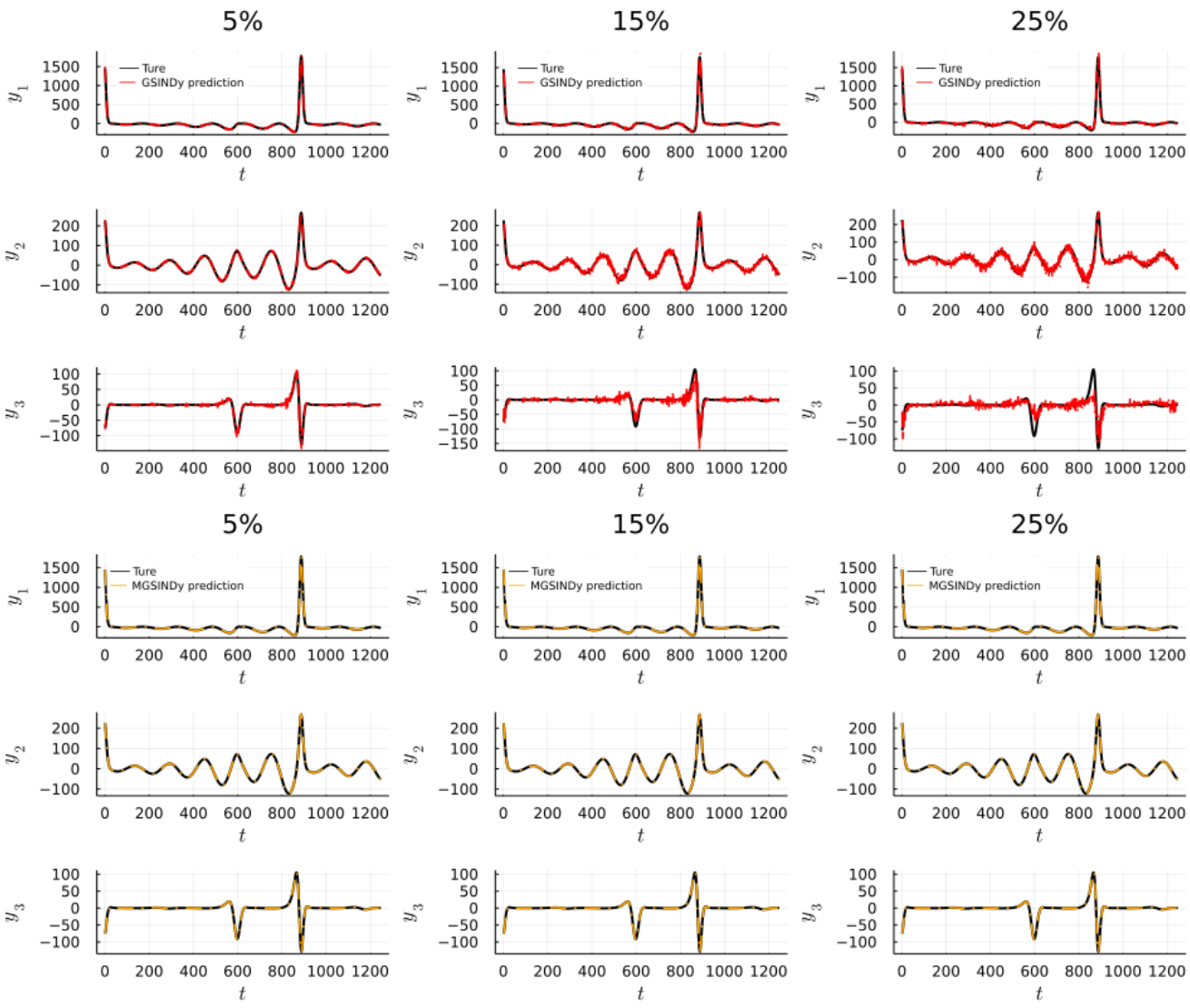

Fig. 4: Graphical illustration of GSINDy and MGSINDy output predictions for Rössler attractor example at different noise levels using clean data as references.

where **a** represents a set of arbitrary numbers and each value inside **a** corresponds to a rational model term. This type of library is not recommended in the absence of precise prior system knowledge.

Alternatively, equation (25) can be rearranged into

$$\dot{x} = 0.6 - 3x - \frac{10}{3}x\dot{x}. \tag{27}$$

In this case, the nonlinear ODE is converted into a linear-in-parameter form, and the standard second-order polynomial library is sufficient by assigning $\dot{x}$ as the second variable. Then, the input vector is augmented as, $\mathbf{x} = \begin{bmatrix} x & \dot{x} \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$. When creating the second-order polynomial library, individual $\dot{x}$ term is removed from the library to avoid the trivial solution.

The output model is constructed in the following manner:

$$y = 3\dot{x} + 2\dot{x}^2. \tag{28}$$

During simulation, equation (25) is solved using the python function *solve ivp* with dt = 0.001 and T = 4 to generate the clean input data. Following that, three levels of Gaussian noise of 10%, 20%, and 30%, are applied to the clean data. We chose $\lambda_x = 0.3$ for SINDy and $\lambda_y = 0.5$ for GSINDy, and the same parameters are assigned for MGSINDy. Three optimization loops were applied for MGSINDy optimization process.

Table 3 presents the GSINDy and MGSINDy predictions' ARMSE values over 1000 Monte Carlo simulation runs for the Michaelis-Menten kinetics example. According to the ARMSE value comparison, denoising the input data enables the MGSINDy to identify a more accurate output model, resulting in more accurate output predictions and demonstrating that MGSINDy is a noise-robust system identification technique.

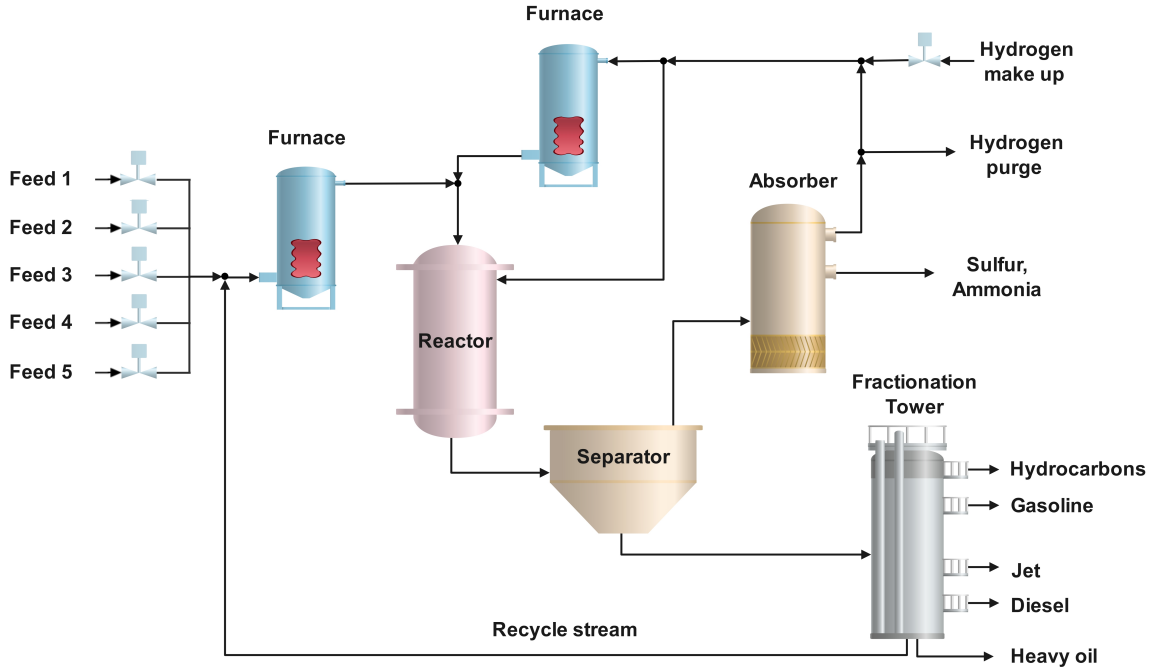

Fig. 5: Graphical illustration of the DHT unit.

Table 3
ARMSE from the GSINDy and MGSINDy for Michaelis-Menten Kinetics Analysis.

Noise level(%)	Variable	GSINDy	MGSINDy
10%	y	0.022	3.4e-3
20%	y	0.048	4.5e-3
30%	y	0.070	5.7e-3

5.3. Diesel Hydrotreating Unit Digital Twin Construction

In petroleum industry, a DHT unit is critical to desulfurizing the crude oil input to comply with applicable environmental criteria for generating clean fuels. A digital twin for the DHT unit can assist the producer in optimizing process control, allowing them to overcome the double-edged challenge of achieving the ever-stricter diesel fuel regulations while simultaneously generating more diesel products from lower-quality feedstocks.

Fig. 5 presents a simplified diesel hydrotreating process. Seven input variables are utilized in the digital twin modelling of this DHT unit, including flowrates of five feed streams and one recycling stream, as well as the reactor temperature. For proprietary reasons, specifics concerning the input process variables have been withheld, and all the data are normalized in this case study. In this process, five feed streams are mixed together with the recycling stream and fed into the furnace to be heated to the reactor operating temperature. Meanwhile, a portion of the inlet hydrogen is directly get into the reactor, while the remainder is heated through a furnace and combined with the heated input stream to enter the catalytic reactor. The diesel hydrotreating process occurs inside the reactor and produces light and heavy reaction products. The output from the reactor is cooled before being transported to a separator. The light reaction products separated from the separator will get into an absorber, which will remove sulphur and ammonia from the hydrogen. The heavy reaction products will enter the fractionation tower for ultimate separation, yielding hydrocarbons, gasoline, jet, diesel, and heavy oil, (Gary et al., 2007; Ahmad et al., 2011).

Flowrates of three critical outputs are selected as the objective output variables, including gasoline, jet, and diesel yields, all expressed in barrels per hour (BPH). Since the number of input and output variables are different, GSINDy is applied to provide the benchmark for this case study. According to hydrotreater reaction kinetics, the first-principles term, $e^{-\frac{E_a}{RT}}$, is added to the model term libraries, where T is the reactor temperature in unit of Kelvin and is one of the input variables, and E_a represents the activation energy. The value of E_a varies based on the feed streams composition, types of catalyst, and reaction conditions. In this case study, $E_a = 121.4$ KJ/mol is utilized for analysis and the gas-law constant R equals 8.314 J/molK (Robinson and Dolbear, 2006). The values of this first-principles term are normalized inside the library.

Subsequently, customized model term libraries are constructed for this DHT unit digital twin identification. The model term libraries for input time-series dynamics and the output predictions are

$$\boldsymbol{\Theta}_{\mathbf{x}}(\mathbf{X}) = \left[\mathbf{X} \quad e^{-\frac{E_a}{RX_7}}\right], \tag{29}$$

$$\boldsymbol{\Theta}_{\mathbf{y}}(\mathbf{X}) = \begin{bmatrix} \mathbf{X} & e^{-\frac{E_a}{RX_7}} & \tanh(3.2 \cdot X_{1-6}) \end{bmatrix}, \tag{30}$$

where $\mathbf{X} = \begin{bmatrix} X_1 & \dots & X_7 \end{bmatrix}$, and X_{1-6} represents the six flowrates of the feed and recycle streams and X_7 is the reactor temperature. The hyperbolic tangent of the six flowrates, $\tanh(3.2 \cdot X_{1-6})$, are added to the output model term library to help capture the nonlinearity, and 3.2 is a constant scale factor scaling the X_{1-6} values within the tanh operator.

The thresholding parameters for input time-series dynamics and output model identifications are $\lambda_x = 0.2$, $\lambda_y = 0.22$, respectively. In MGSINDy, $dt = 0.01$ is used to denoise the input measurement. The Adam optimizer was used with two optimization loops. In this example, hourly data over one year are available. After removing outliers, 8593 samples are used to construct the digital twin model, while the remaining 3682 samples are used to assess the models' performance. Different from the numerical examples, which apply the identified output models on denoised data set, the MGSINDy-identified models are directly applied to the noisy test data for performance testing in this case study.

Table 4 summarizes the prediction accuracy based on testing samples in terms of mean squared error (MSE). According to the MSE comparison, by denoisng the input measurements, the MGSINDy is able to construct more accurate digital twin models for the DHT unit output predictions, yielding 18% to 41% performance improvement. Frequently, SINDy-based algorithms focus on identifying parsimonious governing equations, which could provide more interpretability and larger domain of applicability. However, after implementation, the process dynamics may change along time. Therefore, it would be beneficial to monitor the performance of the implemented digital twin model and adjust the model online when necessary. Several SINDy-based online model updating frameworks have been proposed in literature (Bhadriraju, Narasingam, et al., 2019; Quade et al., 2018). In the presence of measurement noise, the MGSINDy can be seamlessly integrated with these frameworks by replacing the SINDy-identified models with the MGSINDy-identified models to provide a noise-robust, online-adjustable digital twin identification framework.

Table 4
MSE from the GSINDy and MGSINDy for DHT Unit Objective Outputs Prediction.

Objective Outputs (BPH)	GSINDy	MGSINDy	% Performance Increase
Gasoline	0.53	0.37	30
Diesel	0.17	0.14	18
Jet	0.39	0.23	41

6. Conclusions

Digital twin is the vital concept that enables the use of digital technology to boost system operational efficiency, hence increasing economic profit and lowering labour costs. Numerous conceptual and structural introductions have been introduced to demonstrate the advantages of digital twins. However, systematic digital twin identification approaches that can automatically discover digital twin models in the presence of input measurement noise have not been well analyzed. In this work, the MGSINDy is proposed as an extension of the SINDy (Brunton et al., 2016), the GSINDy (J Wang et al., 2022), and the MSINDy (Kaheman et al., 2020). The proposed MGSINDy algorithm simultaneously identifies the input time-series dynamics and the output model, while denoising the input measurements. Comparing to traditional noise filtering approaches, such as low-pass filtering, which only consider the information of input, the proposed MGSINDy algorithm utilizes both input time-series dynamics and the output information to denoise the input, resulting in a more prediction-beneficial denoising procedure. Both numerical examples and industrial case study demonstrate the effectiveness of the proposed approach. In each numerical example, the capacities of denoising input data and identifying the digital twin model are tested against three distinct levels of noise. In the industrial DHT unit case study, the MGSINDy successfully improves the output prediction accuracy for 18% to 41%, indicating higher accuracy of digital twin identification. The proposed modelling algorithm can be applied to general industrial process digital twin constructions, such as paper production, automobile manufacturing, and solvent recovery to automatically discover the digital twin models in the presence of input measurement noise. Additionally, the proposed MGSINDy algorithm can be integrated with the SINDy-based online model updating frameworks to perform online performance monitoring and model adjusting.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Jingyi Wang: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Jesús Moreira**: Data curation, Resources, Supervision. **Yankai Cao**: Conceptualization, Methodology, Resources, Supervision, Writing - review & editing. **R. Bhushan Gopaluni**: Conceptualization, Funding acquisition, Methodology, Resources, Supervision, Writing – review & editing.

Acknowledgement

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

References

Abdullah F, Wu Z, and Christofides P D, Feb. 2021. Data-based reduced-order modeling of nonlinear two-time-scale processes. Chemical Engineering Research and Design; 166,1–9. DOI: 10.1016/j.cherd.2020.11.009.

Ahmad M I, Zhang N, and Jobson M, 2011. Integrated design of diesel hydrotreating processes. Chem. Eng. Res. Des. 89, (7):1025–1036. DOI: 10.1016/j.cherd.2010.11.021.

Aversano G et al., 2019. Application of reduced-order models based on PCA &amp Kriging for the development of digital twins of reacting flow applications. Comput. Chem. Eng. 121,422–441. DOI: 10.1016/j.compchemeng.2018.09.022.

Bangi M S F and Kwon J S.-I, Jan. 2023. Deep hybrid model-based predictive control with guarantees on domain of applicability. AIChE Journal; DOI: 10.1002/aic.18012.

Bergman T L et al., 2011. Introduction to heat transfer. John Wiley & Sons.

Bhadriraju B, Bangi M S F, et al., Sept. 2020. Operable adaptive sparse identification of systems: Application to chemical processes. AIChE Journal; 66, (11). DOI: 10.1002/aic.16980.

Bhadriraju B, Kwon J S.-I, and Khan F, 2021. OASIS-P: Operable Adaptive Sparse Identification of Systems for fault Prognosis of chemical processes. Journal of Process Control; 107,114–126. DOI: 10.1016/j.jprocont.2021.10.006.

Bhadriraju B, Narasingam A, and Kwon J S.-I, 2019. Machine learning-based adaptive model identification of systems: Application to a chemical process. Chem. Eng. Res. Des. 152,372–383. DOI: 10.1016/j.cherd.2019.09.009.

Brunton S L, Proctor J L, and Kutz J N, 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. U.S.A. 113, (15):3932–3937. DOI: 10.1073/pnas.1517384113.

Cavone G et al., 2022. MPC-Based Process Control of Deep Drawing: An Industry 4.0 Case Study in Automotive. IEEE Trans. Autom. Sci. Eng.1–13. DOI: 10.1109/TASE.2022.3177362.

Champion K et al., 2020. A Unified Sparse Optimization Framework to Learn Parsimonious Physics-Informed Models From Data. IEEE Access; 8,169259–169271. DOI: 10.1109/access.2020.3023625.

Champion K P, Brunton S L, and Kutz J N, 2019. Discovery of Nonlinear Multiscale Systems: Sampling Strategies and Embeddings. SIAM J. Appl. Dyn; 18, (1):312–333. DOI: 10.1137/18m1188227.

Dang H V, Tatipamula M, and Nguyen H X, 2022. Cloud-Based Digital Twinning for Structural Health Monitoring Using Deep Learning. IEEE Trans. Industr. Inform. 18, (6):3820–3830. DOI: 10.1109/TII.2021.3115119.

Fasel U et al., 2021. Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. arXiv preprint arXiv:2111.10992; DOI: 10.48550/arXiv.2111.10992.

Gary J H et al., 2007. Petroleum refining: technology and economics. CRC press.

Gaspard P, 2005. Rössler systems. Encyclopedia Nonlinear Sci. 231,808–811.

Hung M.-H et al., 2022. A Novel Implementation Framework of Digital Twins for Intelligent Manufacturing Based on Container Technology and Cloud Manufacturing Services. IEEE Trans. Autom. Sci. Eng.1–17. DOI: 10.1109/TASE.2022.3143832.

Jimenez J I, Jahankhani H, and Kendzierskyj S, 2019. Health Care in the Cyberspace: Medical Cyber-Physical System and Digital Twin Challenges. In: Digital Twin Technol. and Smart Cities. Springer International Publishing:79–92. DOI: 10.1007/978-3-030-18732-3_6.

Kaheman K, Brunton S L, and Kutz J N, 2020. Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data. arXiv preprint arXiv:2009.08810; DOI: 10.48550/arXiv.2009.08810.

Kaiser E, Kutz J N, and Brunton S L, 2018. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. Proc. Math. Phys. Eng. 474, (2219):20180335. DOI: 10.1098/rspa.2018.0335.

Kaur M J, Mishra V P, and Maheshwari P, 2019. The Convergence of Digital Twin, IoT, and Machine Learning: Transforming Data into Action. In: Digital Twin Technol. and Smart Cities. Springer International Publishing:3–17. DOI: 10.1007/978-3-030-18732-3_1.

Kender R et al., 2021. Development of a digital twin for a flexible air separation unit using a pressure-driven simulation approach. Comput. Chem. Eng. 151,107349. DOI: 10.1016/j.compchemeng.2021.107349.

Kuznetsov N, Mokaev T, and Vasilyev P, 2014. Numerical justification of Leonov conjecture on Lyapunov dimension of Rossler attractor. Commun. Nonlinear Sci. Numer. Simul. 19, (4):1027–1034. DOI: 10.1016/j.cnsns.2013.07.026.

Mangan N M et al., 2016. Inferring Biological Networks by Sparse Identification of Nonlinear Dynamics. IEEE Trans. Mol. Biol. Multi-Scale Commun. 2, (1):52–63. DOI: 10.1109/tmbmc.2016.2633265.

Maris D T and Goussis D A, 2015. The “hidden” dynamics of the Rössler attractor. Phys. D: Nonlinear Phenom. 295-296,66–90. DOI: 10.1016/j.physd.2014.12.010.

Narasingam A and Kwon J S.-I, Nov. 2018. Data-driven identification of interpretable reduced-order models using sparse regression. Computers &amp Chemical Engineering; 119,101–111. DOI: 10.1016/j.compchemeng.2018.08.010.

Papanagnou C I, 2019. A Digital Twin Model for Enhancing Performance Measurement in Assembly Lines. In: Digital Twin Technol. and Smart Cities. Springer International Publishing:53–66. DOI: 10.1007/978-3-030-18732-3_4.

Quade M et al., 2018. Sparse identification of nonlinear dynamics for rapid model recovery. Chaos; 28, (6):063116. DOI: 10.1063/1.5027470.

Raviprakash K, Huang B, and Prasad V, 2022. A hybrid modelling approach to model process dynamics by the discovery of a system of partial differential equations. Comput. Chem. Eng.107862. DOI: 10.1016/j.compchemeng.2022.107862.

Robinson P R and Dolbear G E, 2006. Hydrotreating and hydrocracking: fundamentals. Practical advances in petroleum processing; 1,177–217.

Rudy S H, Kutz J N, and Brunton S L, 2019. Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. J. Comput. Phys. 396,483–506. DOI: 10.1016/j.jcp.2019.06.056.

Shardt Y A and Huang B, 2013. Data quality assessment of routine operating data for process identification. Comput. Chem. Eng. 55,19–27. DOI: 10.1016/j.compchemeng.2013.03.029.

Subramanian R, Moar R R, and Singh S, Mar. 2021. White-box Machine learning approaches to identify governing equations for overall dynamics of manufacturing systems: A case study on distillation column. Machine Learning with Applications; 3,100014. DOI: 10.1016/j.mlwa.2020.100014.

Sun W et al., 2021. Adaptive Federated Learning and Digital Twin for Industrial Internet of Things. IEEE Trans. Industr. Inform. 17, (8):5605–5614. DOI: 10.1109/TII.2020.3034674.

Tao F et al., 2018. Digital twin in industry: State-of-the-art. IEEE Trans. Industr. Inform. 15, (4):2405–2415. DOI: 10.1109/TII.2018.2873186.

Tuegel E J et al., 2011. Reengineering Aircraft Structural Life Prediction Using a Digital Twin. Int. J. Aerosp; 2011,1–14. DOI: 10.1155/2011/154798.

Uhlemann T H.-J, Lehmann C, and Steinhilper R, 2017. The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0. Procedia CIRP; 61,335–340. DOI: 10.1016/j.procir.2016.11.152.

Wang J et al., 2022. Time-Variant Digital Twin Modeling through the Kalman-Generalized Sparse Identification of Nonlinear Dynamics. In: Proc. Am. Control. Conf. IEEE.

Wang S et al., 2016. Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. Comput. Netw. 101,158–168. DOI: 10.1016/j.comnet.2015.12.017.

Wang T et al., 2022. Adaptive Optimization Method in Digital Twin Conveyor Systems via Range-Inspection Control. IEEE Trans. Autom. Sci. Eng. 19, (2):1296–1304. DOI: 10.1109/TASE.2020.3043393.

Wright L and Davidson S, Mar. 2020. How to tell the difference between a model and a digital twin. Advanced Modeling and Simulation in Engineering Sciences; 7, (1). DOI: 10.1186/s40323-020-00147-4.

Xu F, Huang B, and Tamayo E C, 2008. Performance assessment of MIMO control systems with time-variant disturbance dynamics. Comput. Chem. Eng. 32, (9):2144–2154. DOI: 10.1016/j.compchemeng.2008.02.003.

Yin X, Bo S, et al., 2021. Consensus-based approach for parameter and state estimation of agro-hydrological systems. AIChE Journal; 67, (2):e17096. DOI: 10.1002/aic.17096.

Yin X and Huang B, 2022. Event-Triggered Distributed Moving Horizon State Estimation of Linear Systems. IEEE Trans. Syst. Man Cybern.: Syst. DOI: 10.1109/TSMC.2022.3146182.