

Article

Not peer-reviewed version

---

# Adaptive Digital Twin Identification with Control: Integration of Extended Kalman Filter-Based Recursive Sparse Nonlinear Identification with Model Predictive Control

---

[Jingyi Wang](#)\*, [Liang Cao](#), [Yankai Cao](#), [R. Bhushan Gopaluni](#)

Posted Date: 12 January 2026

doi: 10.20944/preprints202601.0854.v1

Keywords: adaptive algorithms; digital twins; extended Kalman filter; sparse matrices; model predictive control



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# Adaptive Digital Twin Identification with Control: Integration of Extended Kalman Filter-Based Recursive Sparse Nonlinear Identification with Model Predictive Control

Jingyi Wang <sup>1</sup> , Liang Cao <sup>2</sup>, Yankai Cao <sup>1,\*</sup> and R. Bhushan Gopaluni <sup>1,\*</sup>

<sup>1</sup> Department of Chemical and Biological Engineering, University of British Columbia, Vancouver, BC V6T 1Z3, Canada

<sup>2</sup> Department of Chemical Engineering, Massachusetts Institute of Technology, Boston, MA 02139, United States

\* Correspondence: yankai.cao@ubc.ca (Y.C.); bhushan.gopaluni@ubc.ca (R.B.G.)

## Abstract

The adoption of digital twins has revolutionized industrial process simulation, monitoring, and control effectiveness. However, practical implementations of digital twins are hindered by substantial challenges, including extended development time, diminishing model accuracy, and restricted interactive capabilities. Addressing these critical issues, this paper proposes a comprehensive digital twin development framework that integrates digital twin identification, real-time model updating, and advanced process control. The proposed approach first identifies the offline digital twin model through the sparse identification of nonlinear dynamics algorithm, reducing the digital twin development time while maintaining model fidelity. Then, the identified model is updated by the extended Kalman filter to mitigate the problem of diminishing accuracy. Finally, incorporating the latest updated model into the model predictive control facilitates the control inputs optimization and enhances the interactive capacity of digital twins. Through one industrial case study and two simulation examples, the advantages of the proposed algorithm are demonstrated.

**Keywords:** adaptive algorithms; digital twins; extended Kalman filter; sparse matrices; model predictive control

## 1. Introduction

With ever-expanding technological innovation, the concept of digital twins has emerged as a revolutionary paradigm that connects physical and digital domains [1,2]. By continuously collecting and integrating data from various sources, digital twins not only provide an accurate real-time reflection of the physical entity but also facilitate the process control effectiveness. The development of advanced digital service platforms, such as Microsoft Azure and SAP Industrial Internet of Things (IIoT), has provided a solid foundation for the deployment of the digital twin technique, enabling it to be widely applied across industries to enhance process simulation, state prediction, and input optimization [3–5].

Since its original development, the application area of digital twins has expanded from aerospace to information and communications technology (ICT) as well as manufacturing [6,7]. In [8], a digital twin framework for smart manufacturing was proposed using deep reinforcement learning to perform online optimization for reconfiguration planning. Additionally, in [9], the digital twin for a painting robot was developed to enable painting process practices without consuming physical assets. Furthermore, a modular framework was developed in [10] to implement a flexible and cost-effective digital twin for the power system.

A comprehensive industrial digital twin circle is a smooth combination of different aspects, which include the underlying algorithm that supports simulation, prediction, and control optimization, the

front-end 3D visual modeling for user interactions, and the backend system for data acquisition and PLC command dispatch. Among these aspects, robust and accurate models build the foundations for digital-twins to realize their potential to ensure fidelity, synchronization, and effective interactions between the physical and virtual spaces.[11]. As a result, this research focuses on the algorithm perspective of the digital twin to streamline the digital twin's model identification procedure and integrate the digital twin model with advanced process control.

Striking a balance between model accuracy and complexity is of utmost importance in digital twin identification to prevent overfitting and reduce computational cost [12]. A sparse and structured, function-on-function model was proposed in [13] to hierarchically select the significant variables for digital twin model constructions. Moreover, an adaptive, sparse graph learning approach was proposed in [14], using a digital twin model to generate the performance degradation data, achieving accurate remaining useful life (RUL) prediction for rolling element bearings. Among the sparsity-promoting identification algorithms, the sparse identification of nonlinear dynamics (SINDy) introduced in [15], has gathered increased scholarly interest in recent times. It employs a three-step, sparse regression framework to automatically determine the governing equations for dynamic systems, significantly reducing the modeling time while maintaining model fidelity.

Real-time synchronization is another critical aspect in digital twin algorithm developments, essential for maintaining accuracy over extended periods. To achieve this, adaptive identification is required to perform online model updates while simultaneously minimizing computational burdens. The Kalman filter and its variants, renowned for their recursive estimation nature and capacity to handle system uncertainties, have been adapted to estimate parameters within the system model, facilitating the identification and calibration of models in real time. In [16], a Kalman filter and SINDy integrated algorithm was proposed to perform online-adaptive, multi-input, multi-output (MIMO) digital twin identifications, offering a dynamic and responsive solution for digital twin modeling in complex systems. However, this algorithm only addresses the modeling and synchronization aspects of digital twin development, disregarding the interactive functionality of digital twins [17].

To facilitate effective interactions between user directives and system operations, it is imperative to integrate advanced process control algorithms into the development of digital twins. The model predictive control (MPC) plays a significant role in the field of advanced process control and has been applied extensively in various industries, including automobile controlling [18] and urban system controlling [19]. By incorporating external inputs into SINDy's framework, the SINDy with control (SINDYc) algorithm was introduced in [20]. This algorithm was then integrated with the MPC to establish a dependable framework for concurrently realizing the modeling and interaction functionalities of digital twin algorithms. However, based on a time-invariant identification approach, this integrated algorithm may experience a decline in prediction accuracy when the system undergoes dynamic changes, thereby leading to suboptimal control performance.

Overall, existing digital twin development algorithms typically address one or two of the fundamental aspects—modeling, synchronization, and interaction. The motivation of this study is to propose an algorithm that concurrently minimizes the initial modeling time and performs online model updating while ensuring the model fidelity and effective interactions between physical and virtual environments [12,21]. To achieve this, this paper integrates SINDYc's framework with EKF's model parameter calibration to standardize the digital twin modeling and updating procedure while applying MPC-based control actions to promote effective interactions.

The major contributions of this paper are summarized as follows.

1. The SINDYc-based, three-step framework is utilized to identify the initial digital twin models, considering both first-principles knowledge and data-driven techniques. Implementing such a standardized, automatic identification algorithm reduces the digital twin algorithm development time while enhancing model fidelity.

2. The integration of the EKF with SINDYc's framework enables online digital twin model updating according to real-time measurements and prevents diminishing synchronization accuracy over time.
3. The proposed EKF-based, recursive, sparse nonlinear identification is further integrated with the MPC, using the latest updated digital twin model to provide state predictions over the prediction horizon. This integration facilitates control inputs optimization while promoting effective interactions between user instructions and the system's dynamic behavior.

## 2. Preliminaries

### 2.1. Extended Kalman Filter

Consider a nonlinear system with control input:

$$x_t = f_{t-1}(x_{t-1}, u_{t-1}) + w_{t-1}, \quad (1)$$

$$y_t = h_t(x_t) + v_t, \quad (2)$$

where  $x$  denotes the system state,  $u$  represents the control input,  $y$  indicates the output measurement, and  $t$  corresponds to the time instant. Additionally, the process noise,  $w$ , and the measurement noise,  $v$ , are considered and assumed to be zero mean, uncorrelated Gaussian white noise, with  $Q_{KF}$  and  $R_{KF}$  representing their respective covariance matrices. The function,  $f$ , captures the nonlinear relationship between the current state and the previous state and control input, while  $h$  denotes the nonlinear relationship between the system state and the measurement.

The EKF adheres to a two-step process comprising prediction and correction [22]. The prediction step can be described as follows:

$$\hat{x}_t^- = f_{t-1}(\hat{x}_{t-1}, u_{t-1}), \quad (3)$$

$$P_t^- = F_{t-1}P_{t-1}F_{t-1}^T + Q_{KF}, \quad (4)$$

where  $\hat{x}^-$  and  $\hat{x}$  are prior and posterior state estimates, respectively. Correspondingly,  $P^-$  and  $P$  represent the prior and posterior state estimation error covariance matrices. Notably,  $F$  denotes the Jacobian matrix of the state model, and it is computed by linearizing the nonlinear state model,  $f$ .

In the correction steps, first, the Kalman gain matrix,  $K$ , is computed as

$$K_t = P_t^- H_t^T (H_t P_t^- H_t^T + R_{KF})^{-1}, \quad (5)$$

where  $H$  stands for the Jacobian matrix corresponding to the nonlinear output model,  $h$ . Subsequently, the innovation sequence is computed by  $y_t - h_t(\hat{x}_t^-)$ . Finally, the posterior state estimate and the posterior state estimation error covariance are derived as

$$\hat{x}_t = \hat{x}_t^- + K_t(y_t - h_t(\hat{x}_t^-)), \quad (6)$$

$$P_t = (I - K_t H_t) P_t^-. \quad (7)$$

Besides state estimation, the EKF has also been employed to perform recursive parameter estimation through state vector augmentation to include the parameters as additional states. Through this augmentation, the EKF simultaneously updates both the original system states and the model parameters based on new measurements. The dual estimation capability makes the EKF an ideal tool for performing online digital twin model updating.

$$J = \sum_{t=0}^{N_p-1} \left( (\hat{x}_t - r_t)^T Q (\hat{x}_t - r_t) + \hat{u}_t^T R_u \hat{u}_t \right) + \underbrace{\left( \hat{x}_{N_p} - r_{N_p} \right)^T Q_{N_p} \left( \hat{x}_{N_p} - r_{N_p} \right)}_{\text{terminal cost}} \quad (8)$$

## 2.2. Model Predictive Control

The detailed procedure of MPC is illustrated in Figure 1. At a specific time instant  $j$ , several components are defined: historical states, depicted as a red dotted line, and past control inputs, represented by a black solid line. The green dashed line signifies the reference trajectory that the system state is intended to follow. Central to MPC is the determination of the control sequence,  $u(\cdot|x_j) = \{u_{j+1} \dots u_{j+N_c}\}$ , based on the current state,  $x_j$ . This sequence spans the control horizon,  $T_c = N_c \Delta T$ , where  $N_c$  is the number of control intervals and  $\Delta T$  is the model sampling interval. This control sequence is determined by solving an optimization problem that minimizes the cost function,  $J$ , over the prediction horizon,  $T_p = N_p \Delta T$ , with  $N_p$  representing the number of prediction intervals.

It is notable that  $\Delta T$ , the model sampling interval may differ from the measurement sampling interval. Typically, the control horizon,  $T_c$ , is equal to the prediction horizon,  $T_p$ . As a result, in this study, the condition that  $N_c = N_p$  is considered. Although the control sequence over the entire control horizon is determined, only the immediate next control input,  $u_{j+1}$ , is implemented. Subsequently, the moving horizon window advances by one model sampling interval, leading to a reinitialization and repetition of the optimization process [23–25].

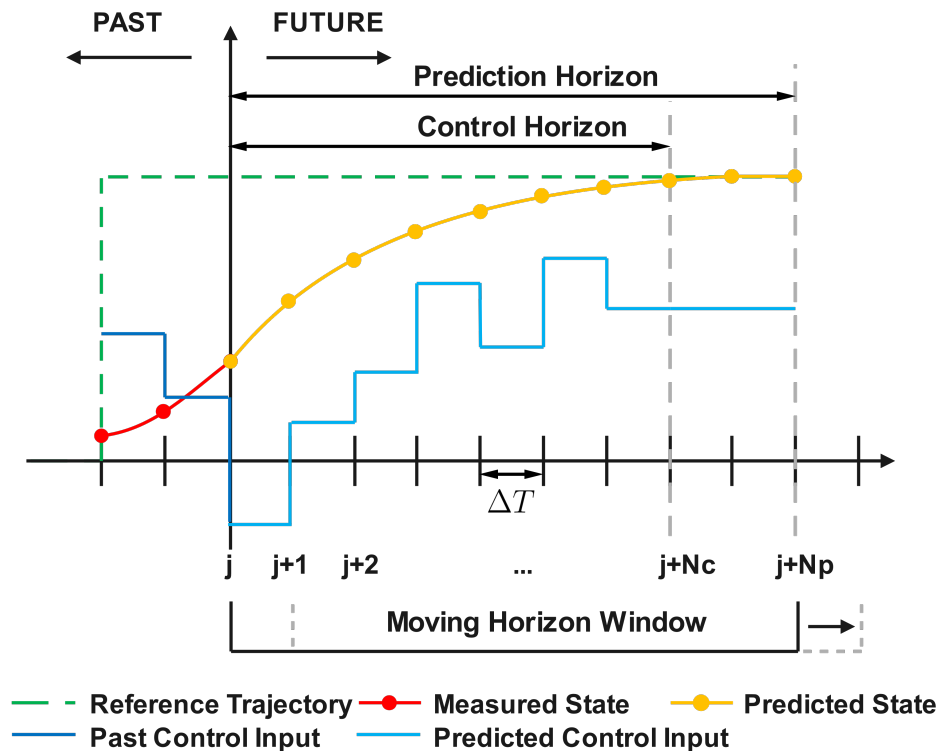


Figure 1. Graphical illustration of MPC.

When solving the optimization problem, standing at time instant zero, the cost function  $J$ , shown in (8), is subject to the system dynamics and the input constraints. In (8),  $r$  represents the reference trajectory and  $\hat{u}$  is the optimized control input, while  $Q$ ,  $Q_{N_p}$  and  $R_u$  are weight matrices to perform penalization.

### 3. Problem Statement

The effectiveness of a digital twin and the performance of the MPC are intrinsically linked to the accuracy and robustness of the underlying model. First-principles models demand extensive time and process expertise to develop but are not robust under system disturbances. In contrast, data-driven models are less time-consuming to develop, but their performance heavily depends on the data quality and is unstable when the process dynamic changes. The SINDYc addresses these challenges through the development of a comprehensive feature library incorporating both first-principles knowledge and data-driven techniques [20]. The detailed feature library construction procedure is introduced in Section 4.1.

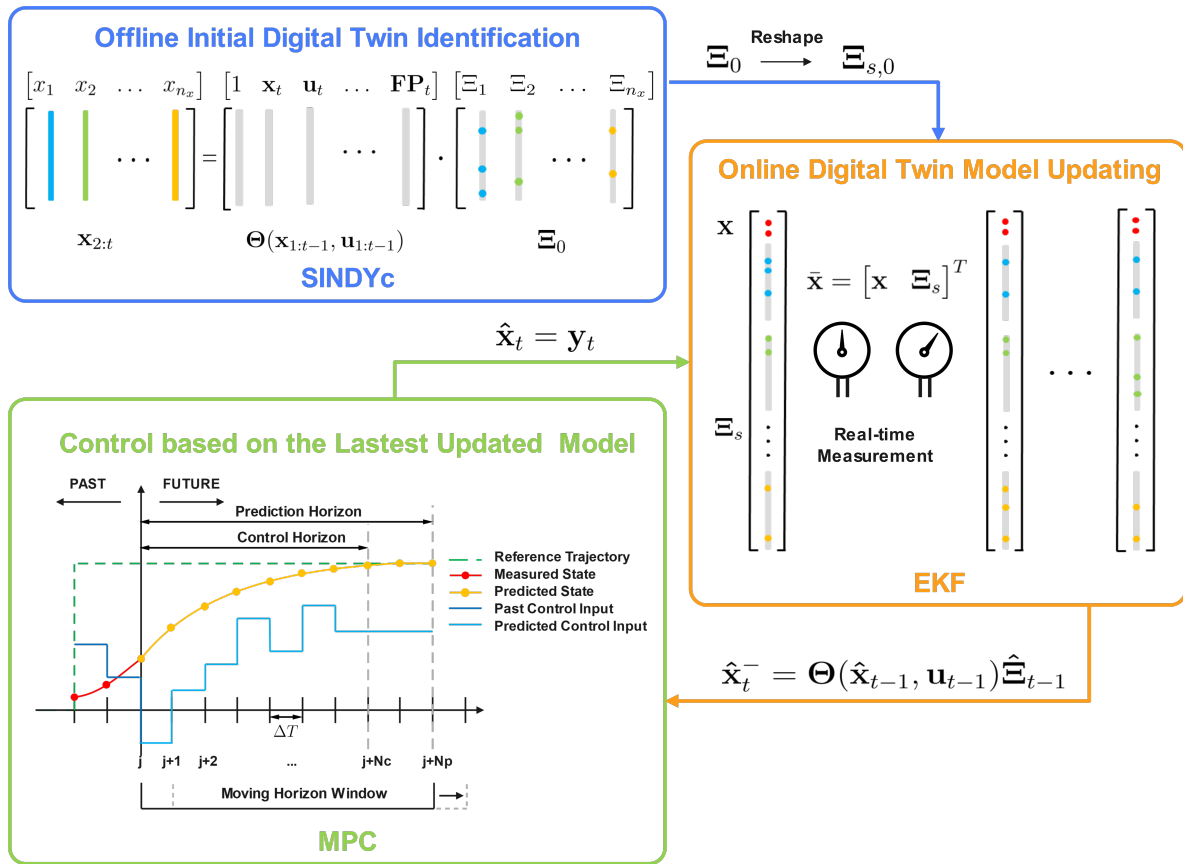
Considering (1) and (2), when applying SINDYc to identify nonlinear state dynamics,  $f$ , a feature library containing control input,  $\Theta(\mathbf{x}, \mathbf{u})$ , is first constructed. Subsequently, a sparse regression problem is solved to determine the necessary features required to construct the digital twin model. The problem is formulated as follows:

$$\mathbf{x}_t = \Theta(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \cdot \Xi_{t-1}, \quad (9)$$

where  $\mathbf{x} = [x_1 \ \dots \ x_{n_x}]$  and  $\mathbf{u} = [u_1 \ \dots \ u_{n_u}]$  with  $n_x$  and  $n_u$  denoting the respective numbers of states and control inputs. The matrix,  $\Xi \in \mathbb{R}^{n_\Theta \times n_x}$ , is defined as the sparse parameter matrix, and  $n_\Theta$  is the number of features within the feature library.

To promote model sparsity while maintaining accuracy, an iterative sequential least squares regression algorithm is implemented in SINDYc, incorporating a thresholding parameter,  $\lambda$ . Parameters whose magnitudes are below this threshold are set to zero. Subsequently, another regression step is performed on the remaining active features. This iterative process leads to convergence, resulting in a sparse parameter matrix,  $\Xi$ . Additional information about solving the sparse regression problem can be found in [15,16,20].

Nevertheless, the model identified by SINDYc is inherently time-invariant. Given the ever-evolving operational conditions within dynamic systems, an effective digital twin identification approach should possess the capability for online adaptability to accommodate the continuous changes in system dynamics. Therefore, a time-variant digital twin model is mandatory to continuously synchronize with the most up-to-date conditions of the system, facilitating optimal control strategy determinations. Consequently, the objective of this research is to first identify the initial digital twin model through the SINDYc algorithm and then update the model feature selections and the corresponding parameters through integration with the EKF. The SINDYc and EKF integrated recursive digital twin identification algorithm is finally integrated with MPC to enable control inputs optimization, promoting interactive capacities of digital twins. A schematic diagram of the proposed framework is shown in Figure 2.



**Figure 2.** Schematic diagram of integrating the EKF-based, recursive, sparse nonlinear digital twin identification with MPC.

#### 4. Integration of Extended Kalman Filter-Based Recursive Sparse Nonlinear Digital Twin Identification with Model Predictive Control

In this research, the state vector of the EKF is augmented to encompass both system states and model parameters. The modified EKF is integrated with SINDYc's framework to recursively update the digital twin model according to real-time measurements. Furthermore, to enhance model sparsity, an iterative correction procedure is incorporated within EKF's correction step. When integrating the proposed recursive digital twin identification with MPC, the latest model is utilized to generate predictions of system states over the prediction horizon. Assuming full-state measurements, after implementing the first element of the MPC-determined optimal control sequence, the received measurement is utilized as the initial estimate of the system state in the next model sampling instant.

##### 4.1. Feature Library Construction and Initial Identification

In the initialization phase of the proposed algorithm, historical data,  $\mathbf{x}_{1:t}$  and  $\mathbf{u}_{1:t}$  are collected, where the control input,  $\mathbf{u}$ , is assumed to be able to excite the system. Subsequently, a feature library can be established in the following format,

$$\Theta(\mathbf{x}, \mathbf{u}) = \left[ 1 \quad \mathbf{x} \quad \mathbf{u} \quad \mathbf{x} \otimes \mathbf{x} \quad \mathbf{x} \otimes \mathbf{u} \quad \dots \quad \mathbf{FP} \right], \quad (10)$$

where the symbol  $\otimes$  indicates the product combination among all the variables within the two specified vectors. In addition to the polynomial combinations, more data-driven features could be included, such as trigonometric terms,  $\sin(\mathbf{x})$ ,  $\tanh(\mathbf{x})$ ,  $\cos(\mathbf{x} \otimes \mathbf{u})$ , and the sigmoid function,  $\frac{1}{1+e^{-\mathbf{x}}}$ . Meanwhile, FP signifies the terms generated based on the first-principles information. For example,

when modeling the heat transfer of the humid air flow passing through a desiccant wheel, the heat adsorption of the desiccant material is described by

$$Q = h_v(1 + 0.2843e^{-10.28W}), \quad (11)$$

where  $Q$  represents the heat adsorbed by the regular density silica gel (desiccant material of the wheel),  $h_v$  is the evaporation latent heat of water, and  $W$  is a process variable indicating the water content of the desiccant material. As a result, when identify the digital twin model of a rotary desiccant dehumidifier, the feature,  $e^{-10.28W}$ , can be included in the feature library of the SINDy-based algorithm.

The construction of a feature library usually starts with the inclusion of simple polynomial features and first-principles features. Subsequently, the library complexity is gradually increased if the existing features are insufficient to achieve satisfactory performance. To prevent overfitting, when determining the initial nonlinear state model,  $f$ , a sparse regression problem between  $\mathbf{x}_{2:t}$  and  $\Theta(\mathbf{x}_{1:t-1}, \mathbf{u}_{1:t-1})$ , is solved, generating the initial sparse parameter matrix,  $\Xi_0$ . Later, during the EKF-based online digital twin model updating, a sparsity-promoting correction step is incorporated, which will be further discussed in Section 4.3.

#### 4.2. Extended Kalman Filter Vectors Augmentation

To facilitate the recursive, simultaneous estimation of system states and digital twin model parameters, the proposed algorithm augments the state vector within the EKF algorithm as,  $\bar{\mathbf{x}} = \begin{bmatrix} \mathbf{x} & \Xi_s \end{bmatrix}^T$ . When constructing the one-dimensional  $\bar{\mathbf{x}}$ , the two-dimensional  $\Xi$ , is reshaped into  $\Xi_s$ . Specifically,

$$\Xi_s = \begin{bmatrix} \Xi_1 & \Xi_2 & \dots & \Xi_{n_x} \end{bmatrix}^T. \quad (12)$$

In (12), for  $i = 1$  to  $n_x$ ,  $\Xi_i$  denotes the parameter vector corresponding to each individual state variable. In specific,

$$\Xi_i = \begin{bmatrix} \zeta_{i,1} & \zeta_{i,2} & \dots & \zeta_{i,n_\Theta} \end{bmatrix}, \quad (13)$$

where  $\zeta$  represents the model parameter corresponding to each feature for a specific system state. Furthermore,  $n_\Xi = n_x \times n_\Theta$ , representing the total number of elements within the sparse parameter matrix.

In this context, the state prediction step of the EKF is modified as follows:

$$\begin{aligned} \hat{\mathbf{x}}_t^- &= \begin{bmatrix} f_{t-1}(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}) \\ \mathbf{I} \cdot \hat{\Xi}_{s,t-1} \end{bmatrix} = \begin{bmatrix} \Theta(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}) \hat{\Xi}_{t-1} \\ \hat{\Xi}_{s,t-1} \end{bmatrix} \\ &= \begin{bmatrix} \hat{\mathbf{x}}_t^- & \hat{\Xi}_{s,t}^- \end{bmatrix}^T, \end{aligned} \quad (14)$$

where  $\hat{\mathbf{x}}_t^- \in \mathbb{R}^{n_s}$  represents the augmented prior state estimate and  $n_s = n_x + n_\Xi$ . In this step, the random walk model is employed to predict the digital twin model parameters at the next model sampling instant. Consequently, the prior model parameter estimate,  $\hat{\Xi}_{s,t}^-$ , remains equal to the posterior model parameter estimate from the last model sampling instant.

Accordingly, the prediction of the state estimation error covariance matrix is augmented as

$$\bar{P}_t^- = \bar{F}_{t-1} \bar{P}_{t-1} \bar{F}_{t-1}^T + \bar{Q}_{KF}, \quad (15)$$

where  $\bar{P}_t^- \in \mathbb{R}^{n_s \times n_s}$ , and  $\bar{Q}_{KF} \in \mathbb{R}^{n_s \times n_s}$  denoting the augmented process noise covariance matrix. Additionally,  $\bar{F}$  signifies the augmented Jacobian matrix of the nonlinear state model,

$$\bar{F}_{t-1} = \begin{bmatrix} \frac{\partial f}{\partial \mathbf{x}} \Big|_{\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}} & \frac{\partial f}{\partial \bar{\boldsymbol{\Xi}}_s} \Big|_{\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}_{n_s \times n_s}, \quad (16)$$

where

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{x}} \Big|_{\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}} &= \frac{\partial(\boldsymbol{\Theta}(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}) \hat{\boldsymbol{\Xi}}_{t-1})}{\partial \mathbf{x}} \Big|_{\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}} \\ &= \frac{\partial(\boldsymbol{\Theta}(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}))}{\partial \mathbf{x}} \hat{\boldsymbol{\Xi}}_{t-1}, \end{aligned} \quad (17)$$

and

$$\begin{aligned} \frac{\partial f}{\partial \bar{\boldsymbol{\Xi}}_s} \Big|_{\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}} &= \frac{\partial(\boldsymbol{\Theta}(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}) \hat{\boldsymbol{\Xi}}_{t-1})}{\partial \bar{\boldsymbol{\Xi}}_s} \Big|_{\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}} \\ &= \boldsymbol{\Theta}(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}). \end{aligned} \quad (18)$$

#### 4.3. Sparsity-Promoting Correction

In MPC applications, typically, the availability of full-state measurement can be assumed. Under this assumption,  $\mathbf{y}_t = \mathbf{x}_t$  and  $\hat{\mathbf{y}}_t = \hat{\mathbf{x}}_t^-$  are employed in the subsequent derivations, and the state observability and controllability are assumed in this study. During the correction process, the augmented Kalman gain matrix,  $\bar{K}_t$ , is calculated as

$$\bar{K}_t = \bar{P}_t^- \bar{H}_t^T (\bar{H}_t \bar{P}_t^- \bar{H}_t^T + R_{KF})^{-1}, \quad (19)$$

and the augmented Jacobian matrix,  $\bar{H}_t$ , is computed as

$$\bar{H}_t = \begin{bmatrix} \frac{\partial h}{\partial \mathbf{x}} \Big|_{\hat{\mathbf{x}}_t^-} & \frac{\partial h}{\partial \bar{\boldsymbol{\Xi}}_s} \Big|_{\hat{\mathbf{x}}_t^-} \end{bmatrix}_{n_x \times n_s}, \quad (20)$$

where

$$\frac{\partial h}{\partial \mathbf{x}} \Big|_{\hat{\mathbf{x}}_t^-} = \frac{\partial \hat{\mathbf{x}}_t^-}{\partial \hat{\mathbf{x}}_t^-} = \mathbf{1}, \quad (21)$$

and

$$\begin{aligned} \frac{\partial h}{\partial \bar{\boldsymbol{\Xi}}_s} \Big|_{\hat{\mathbf{x}}_t^-} &= \frac{\partial \hat{\mathbf{x}}_t^-}{\partial \hat{\boldsymbol{\Xi}}_{s,t}^-} \Big|_{\hat{\mathbf{x}}_t^-} \\ &= \frac{\partial(\boldsymbol{\Theta}(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}) \hat{\boldsymbol{\Xi}}_{t-1})}{\partial \hat{\boldsymbol{\Xi}}_{s,t-1}} \\ &= \boldsymbol{\Theta}(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}). \end{aligned} \quad (22)$$

Subsequently, the correction step promotes model sparsity through an iterative procedure. The value of the thresholding parameter,  $\lambda$ , is incrementally increased until a noticeable decline in the model performance is observed. Within this context, elements of the prior estimate of the single-dimensional sparse parameter matrix,  $\hat{\boldsymbol{\Xi}}_{s,t}^-$ , that exhibit magnitudes below this threshold are set to zero. Following this, the sparsity-enhanced  $\hat{\boldsymbol{\Xi}}_{s,t}^-$  is reshaped into the two-dimensional sparse parameter matrix,  $\hat{\boldsymbol{\Xi}}_t^-$ , which is then utilized to generate new prior state variable estimates, denoted as  $\hat{\mathbf{x}}_t'^-$ . The new state variable estimates, together with the sparsity-enhanced  $\hat{\boldsymbol{\Xi}}_{s,t}^-$  will form the new augmented state vector,  $\hat{\mathbf{x}}_t'^-$ .

The innovation sequence contributes to the correction of the new augmented state vector,  $\hat{\mathbf{x}}_t'^-$ , generating the posterior augmented state vector,  $\hat{\mathbf{x}}_t$ . The posterior estimate of the augmented state vector then replaces the  $\hat{\mathbf{x}}_t'^-$  and is subject to another sparsity-promoting correction iteration. Typically,

convergence is achievable through five to ten iterations. Therefore, the iteration parameter,  $k$ , can be set to ten for effective convergence. The overall sparsity-promoting correction procedure can be formulated as follows:

$$\begin{aligned}
 & \mathbf{for} \quad q = 1 : k, \\
 & \quad \left| \hat{\Xi}_{s,t}^- \right| < \lambda = 0, \\
 & \quad \hat{\mathbf{x}}_t'^- = \Theta(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}) \hat{\Xi}_t^-, \\
 & \quad \hat{\mathbf{y}}_t = \hat{\mathbf{x}}_t'^-, \\
 & \quad \hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t'^- + \bar{K}_t(\mathbf{y}_t - \hat{\mathbf{y}}_t), \\
 & \quad \hat{\mathbf{x}}_t'^- = \hat{\mathbf{x}}_t. \\
 & \mathbf{end}
 \end{aligned} \tag{23}$$

Eventually, the sparse parameter matrix,  $\hat{\Xi}_t^-$ , determined from the final iteration, is employed to generate the final posterior estimates of the augmented state vector so as to ensure the model sparsity and the accuracy of posterior state estimates. Furthermore, the augmented posterior state estimation error covariance matrix is

$$\bar{P}_t = (I - \bar{K}_t \bar{H}_t) \bar{P}_t^-. \tag{24}$$

#### 4.4. Integration with Model Predictive Control

The initial identification of  $\Xi_0$  through the SINDYc provides the foundational estimates for the digital twin model parameters. Afterward, the online adjustments of the model features and the corresponding parameters are performed upon receiving the newly acquired measurement at each model sampling instant. To prevent the trivial solution,  $x_t = x_{t-1}$ , the model sampling interval is usually greater than the measurement sampling interval. The real-time data will form the feature library values at each model sampling instant to contribute to the digital twin online modification.

During the MPC execution, weight matrices,  $Q$ ,  $Q_{N_p}$ , and  $R_u$ , are predetermined according to control specifications. In the proposed, integrated, recursive digital twin identification with MPC approach, the latest updated model is utilized to forecast system states over the prediction horizon, where the model remains static. The predicted states are subsequently utilized to compute the optimal control sequence according to (8). Once the initial component of the optimal control sequence has been implemented, full-state measurements,  $\mathbf{y}_t$ , become available. These measurements are utilized in the sparsity-promoting correction procedure to update the system state estimates and sparse model parameters. Following this correction phase, the full-state measurements are then utilized as the initial state estimates,  $\hat{\mathbf{x}}_t$ , for the next model sampling instant. The main steps of the proposed algorithm are shown in Algorithm ?? and the corresponding graphical representation is provided in Figure 3.

It is notable that the SINDy-based identifications are readily applicable to large-scale applications through expanding the comprehensive feature libraries. By emphasizing sparsity promotion, SINDy-based frameworks facilitate the identification of parsimonious digital twin models for large-scale industrial processes. In the context of MPC implementations, the large MIMO control problems are typically decomposed into multiple multi-input, single-output (MISO) problems or simpler MIMO configurations. This decomposition is essential because the complexity of variable interactions in large-scale MPC strategies. Consequently, when solving large-scale control problems, the proposed recursive digital twin identification with control algorithm can be deployed in parallel to efficiently achieve the desired control objectives.

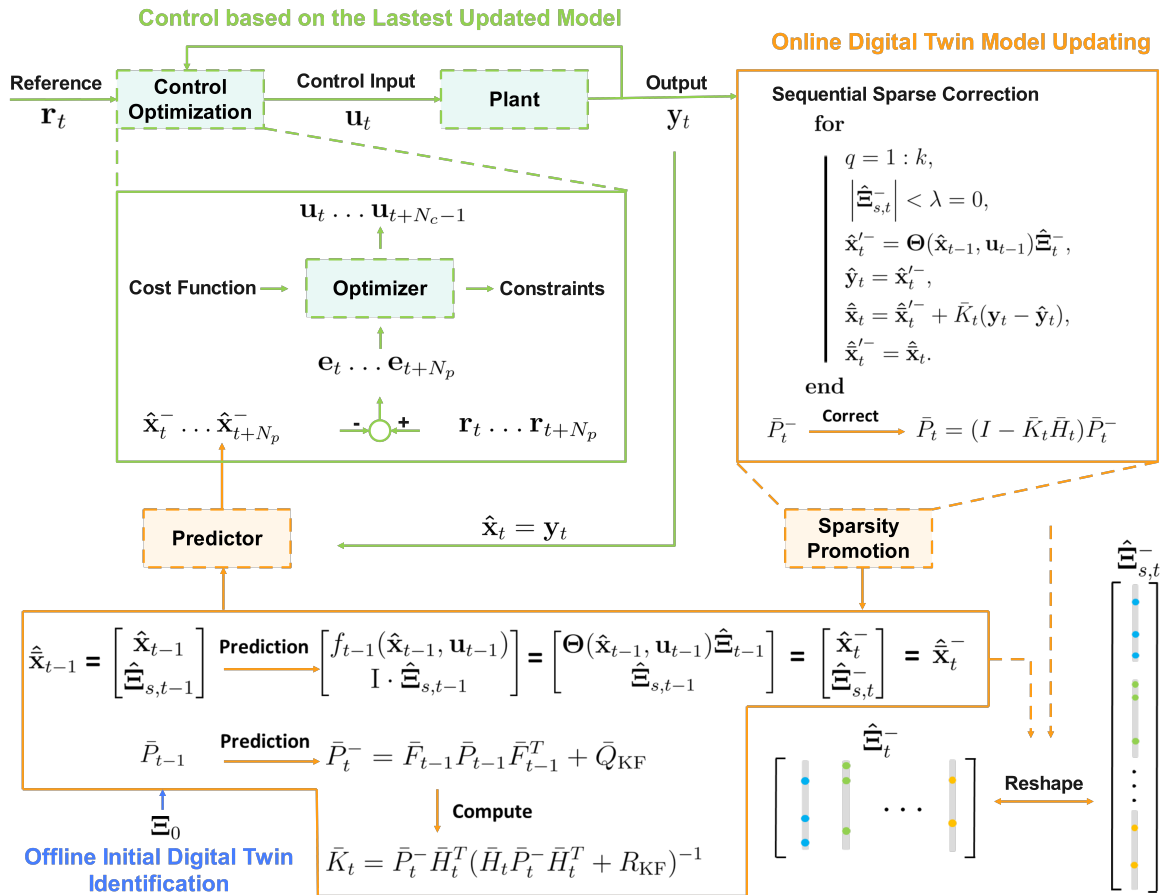


Figure 3. Graphical illustration of integrating the EKF-based, recursive, sparse nonlinear digital twin identification with MPC.

## 5. Case Study

In this section, the prediction accuracy of the proposed EKF-based recursive sparse nonlinear digital-twin identification algorithm is evaluated through an industrial diesel characteristics prediction case study, and the performance is compared with other advanced approaches. Two numerical simulation examples are used to demonstrate the control performance of the proposed recursive digital twin identification with control algorithm against system dynamic changes and training data noise.

### 5.1. Diesel Characteristics Prediction Based on the Digital Twin Model

The diesel hydrotreating (DHT) unit plays a critical role in enhancing fuel performance and meeting regulatory requirements for transportation and industrial applications [26]. Within the DHT unit, the feed streams are initially preheated in the furnace before being fed into the reactor, where key processes such as hydrodesulfurization, hydrodenitrogenation, and hydrocracking occur in the presence of hydrogen and catalysts. The output from the reactor then enters the separator for preliminary separation of lighter and heavier products. The lighter reaction products then proceed to an absorber for sulfur and ammonia removal. The heavier products are further processed in a fractionation tower, where additional separations take place to yield light hydrocarbons, gasoline, jet fuel, diesel, and heavy bottom products [27,28].

The distillation process within the DHT unit ensures that the final diesel product meets the essential quality specifications, such as sulfur content and cetane number. Among these quality parameters, diesel density and diesel cloud point require laboratory measurements, which are often conducted infrequently. To address this limitation, the proposed EKF-based recursive digital twin identification algorithm is used to construct and update the digital twin model, to provide frequent predictions of these two quality values. When evaluating the prediction accuracy, the reference

lab measurements are assumed to be available at three different frequencies. Figure 4 shows the digital twin for the DHT unit. The z-score normalization is applied to the industrial data during the comparative analysis for proprietary reasons.

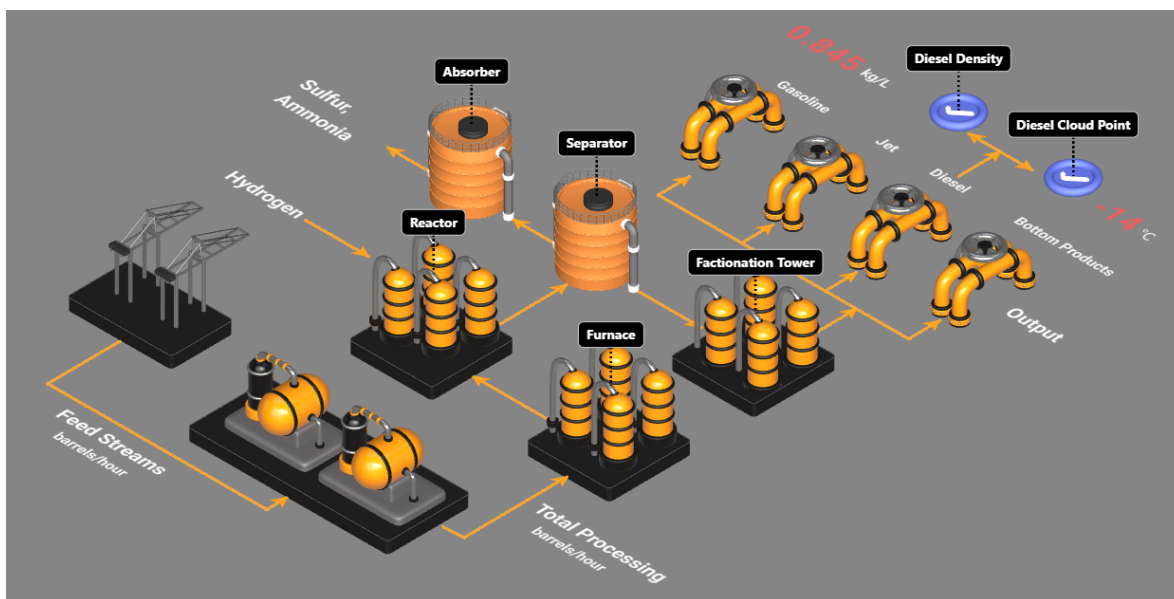


Figure 4. Graphical illustration of the DHT digital twin.

Production hourly data over a ten-month period is used to construct the digital twin model and test the prediction accuracy. To prevent the trivial solution, the model sampling interval is selected as every 24 hours, which gives 295 total samples. In addition to the proposed algorithm, the prediction performance of the neural network with control (NNc), DMD with control (DMDc), and SINDYc is also evaluated to provide a comprehensive comparison. The first 110 samples are used as the training data to identify the time-invariant digital twin model, and the remaining 185 modeling samples are used to test the prediction accuracy of the analyzed approaches.

According to the first-principles knowledge, the reactor inlet temperature plays a significant role in affecting the diesel density and cloud point [29,30]. Consequently, the reactor inlet temperature in the DHT unit is selected as the control input. A second-order polynomial and trigonometric integrated feature library is constructed for both SINDYc and the proposed algorithm,

$$\Theta(\mathbf{x}, \mathbf{u}) = \left[ 1 \quad \mathbf{x} \quad \mathbf{u} \quad \mathbf{x} \otimes \mathbf{x} \quad \sin(\mathbf{x}) \quad \tanh(\mathbf{x}) \right]. \quad (25)$$

In the meanwhile, the thresholding parameter,  $\lambda$ , is set to 0.1.

During the test period, the full-state measurements are assumed to be available at three different model sampling intervals to assess the proposed algorithm's prediction accuracy within the prediction horizon. In addition, when implementing the proposed algorithm,  $\bar{Q}_{KF} = \text{diag}\{0.2 \ 0.4 \ 1e^{-4} \dots 1e^{-4}\}$ ,  $R_{KF} = \text{diag}\{0.4 \ 0.3\}$ , and  $\bar{P}_0 = \text{diag}\{1e^{-4} \dots 1e^{-4}\}$ . Furthermore, a nonlinear autoregressive neural network with external input is used for the NNc, with one hidden layer containing eight neurons to achieve an optimal balance between NNc's model accuracy and complexity.

The prediction mean squared errors (MSEs) of the algorithms considered, based on the 185 test samples, are presented in Table 1. In the right portion of Table 1, the MSE results for the proposed algorithm are provided, with reference lab measurements available at intervals of three, five, and seven days. According to the results presented in Table 1, the DMDc algorithm, which uses linear representations to approximate nonlinear systems, experiences reduced accuracy due to the complex nonlinearities and dynamic behavior changes inherent in the diesel characteristics. In addition, while the NNc algorithm is capable of handling the process nonlinearities, its performance is limited by the

relatively small sample size. Of the three time-invariant algorithms considered, SINDYc outperforms the others, which shows its ability to effectively manage nonlinearities while requiring relatively fewer samples.

Across all three reference measurement availabilities, the proposed algorithm consistently outperforms SINDYc in terms of prediction accuracy. Specifically, the proposed algorithm improves prediction accuracy by 16.3% and 24.9% over SINDYc, even when reference measurements are available only once per week. This significant improvement in prediction accuracy highlights the advantage of integrating the SINDYc with the EKF to maintain the digital twin synchronization precision over time.

**Table 1.** Prediction error comparison in terms of MSE.

| Prediction Objective | Approaches | MSE  | Proposed Algorithm |      |      |
|----------------------|------------|------|--------------------|------|------|
|                      |            |      | Prediction Horizon |      |      |
|                      |            |      | 3                  | 5    | 7    |
| Diesel Density       | NNc        | 0.85 |                    |      |      |
|                      | DMDc       | 0.53 | 0.12               | 0.28 | 0.36 |
|                      | SINDYc     | 0.43 |                    |      |      |
| Cloud Point          | NNc        | 0.87 |                    |      |      |
|                      | DMDc       | 1.48 | 0.29               | 0.42 | 0.48 |
|                      | SINDYc     | 0.64 |                    |      |      |

## 5.2. Simulation Examples

### 5.2.1. The Lotka-Volterra System

The Lotka-Volterra system is a two-dimensional ODE system describing the interactive dynamics between coexisting predator and prey populations within an ecosystem. The dynamics of the prey population,  $x_1$ , and the predator population,  $x_2$ , are mathematically expressed as follows [31]:

$$\dot{x}_1 = \alpha x_1 - \beta x_1 x_2, \quad (26)$$

$$\dot{x}_2 = -\gamma x_2 + \delta x_1 x_2 + u, \quad (27)$$

where  $\alpha$  corresponds to the prey population growth rate in the absence of predators;  $\beta$  characterizes the impact of the predator's presence on the prey's growth rate;  $\gamma$  represents the mortality rate of predators in the absence of their prey; and  $\delta$  signifies the growth rate of the predator when consuming the prey. When implementing the proposed algorithm to perform identification and control for the given ODE system, the discrete-discrete variant method, as outlined in [32], is employed. This approach discretizes the continuous models using the 4th-order Runge-Kutta (RK4) algorithm, facilitating the application of the proposed algorithm.

The system parameter values are altered twice throughout the simulation period. The simulation duration,  $T$ , is set to 100. Since this is a simulated ODE system, the model sampling interval and measurement sampling interval are the same, denoted as  $\Delta t = 0.1$ , resulting in a total of 1000 sampling intervals. For the initial 300 sampling intervals,  $\alpha, \beta, \gamma, \delta$  are established as 0.5, 0.025, 0.5, 0.02 and are adjusted to 0.4, 0.02, 0.4, 0.02 in the next 300 sampling intervals. In the final 400 sampling intervals, the parameters are further modified to 0.3, 0.015, 0.3, 0.03. In addition, the values of control inputs are constrained to  $[-10, 10]$ . The initial states of prey and predator populations are set at 40 and 30, with  $\lambda = 0.01$  and  $N_p = N_c = 6$ . The MPC weight matrices are configured as  $Q = Q_{N_p} = \text{diag}\{0.7 \ 0.7\}$  and  $R_u = 0.7$ . The control objective is to stabilize populations of prey and predator to their respective targeting values,  $\frac{\gamma}{\delta}$  and  $\frac{\alpha}{\beta}$ .

During the evaluation procedure, both MPC-based algorithms and reinforcement learning (RL)-based methods are analyzed for performance comparison. In the MPC implementations, models

identified through the DMDc, NNc, SINDYc, and the proposed EKF-based, recursive, sparse nonlinear identification are utilized. In the RL implementations, the multi-layer perceptron (MLP) policy is employed with two optimization techniques: the on-policy, proximal policy optimization (PPO) and the off-policy, soft actor-critic (SAC).

When implementing the MPC-based algorithms, noise with a magnitude equivalent to 10% of the standard deviation of the clean data is incorporated into the training dataset. Additionally, the initial 100 samples are utilized to identify the time-invariant system models for the DMDc, SINDYc, and the proposed algorithm. In contrast, the NNc requires 400 training samples to achieve stable control performance. The control input used to excite the system for training is

$$u = (2 \sin(t) + \sin(0.1t))^2. \quad (28)$$

An identical polynomial feature library is constructed for both SINDYc and the proposed algorithm,

$$\Theta(\mathbf{x}, \mathbf{u}) = [\mathbf{1} \quad \mathbf{x} \quad \mathbf{u} \quad \mathbf{x} \otimes \mathbf{x}]. \quad (29)$$

In addition, it takes 21.7, 39.1, 219.7, and 1608.6 seconds for DMDc, SINDYc, the proposed algorithm, and NNc to accomplish the MPC implementations, respectively. The DMDc, being a simplified version of the SINDYc, demands less computational time. In contrast, the NNc requires a larger training sample size and longer computational time. Since the proposed algorithm updates the digital twin model at every model sampling instant, its overall computational time is longer than that of DMDc and SINDYc. However, this time-variant approach is significantly more computationally efficient for real-time implementations compared to the need for recomputing the model using time-invariant approaches at each model sampling step.

In the context of RL implementations, the PPO and SAC algorithms require 100,000 and 50,000 clean samples, respectively, to achieve stable and accurate control performance for the initial control objectives, necessitating the generation of additional samples. The corresponding training durations are 206.4 seconds for PPO and 1,168.8 seconds for SAC. Although the SAC method requires fewer training samples, it is an off-policy method that utilizes replay buffer sampling to optimize the control policy, which results in longer training time. The control performance of these analyzed algorithms is shown graphically in Figure 5, where the proposed algorithm is referred to as the EKF-SINDYc.

As observed in Figure 5 (a) to (d), owing to the system dynamic changes and the noise added to the training dataset, NNc, DMDc, and SINDYc all show deviations in controlling the prey population to its objectives. By incorporating the reference measurements along time, the proposed algorithm successfully controls the prey and predator's populations to their respective targets even under system dynamic changes and initial model inaccuracy.

For the RL-based algorithms, clean data are used. According to Figure 5 (e) and (f), the PPO-optimized policy requires a relatively longer settling time and only achieves stable control results after the first simulation period. In contrast, the SAC-optimized policy is capable of reaching quick and stable control results without exhibiting overshooting or oscillations. However, the performance of both RL-based algorithms strongly depends on the sample volume and quality, as well as the training environment, which limits their ability to adapt to new control objectives after the initial 300 sampling intervals. Consequently, due to their extended training duration and limited adaptability to dynamic changes, RL-based algorithms are less suited for real-time digital twin applications compared to MPC-based algorithms under current computational constraints.

Overall, the proposed algorithm maintains effective control over both prey and predator populations across the three simulation periods. This shows the proposed algorithm's robustness to initial model defects and capability to adaptively modify the digital twin models in response to changes in system dynamics, which achieves more effective control performance and ensures stronger interactive capacities.

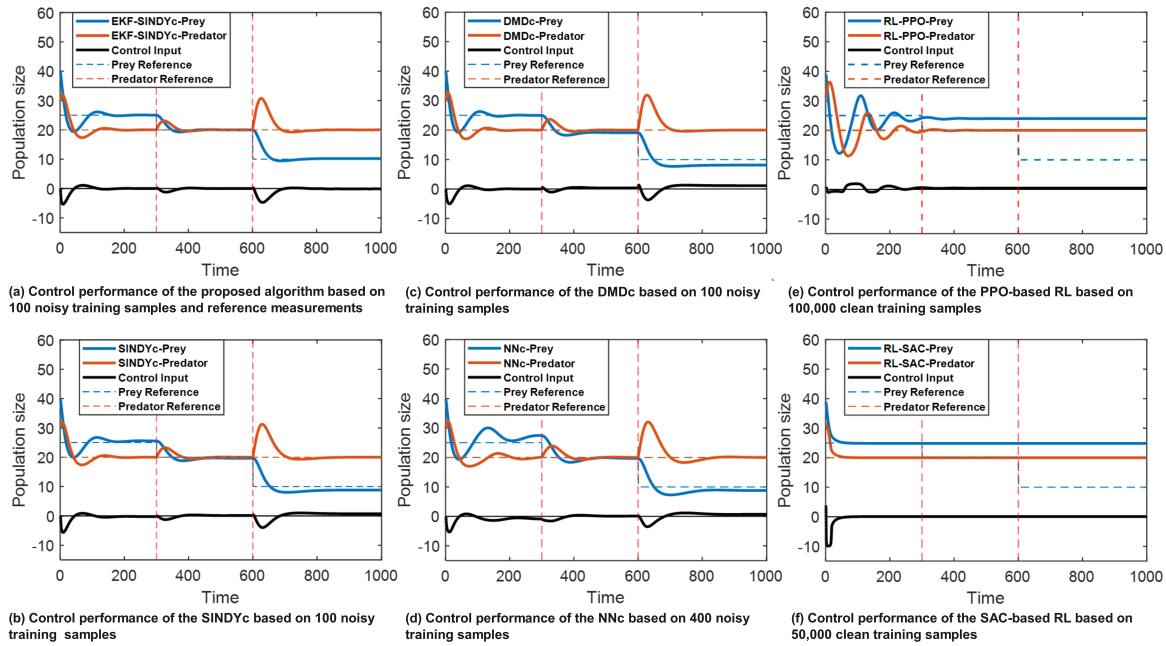


Figure 5. The Lotka-Volterra system control performance comparison.

### 5.2.2. The Discrete-Time System

In the second simulation example, a discrete-time system is given as follows:

$$x_{1,t} = a \tanh(x_{1,t-1}) - b \sin(x_{2,t-1}) + c \cos(x_{1,t-1}) + du_{t-1}, \quad (30)$$

$$x_{2,t} = e \cos(x_{1,t-1}) + f \sin(x_{2,t-1}) + gu_{t-1}, \quad (31)$$

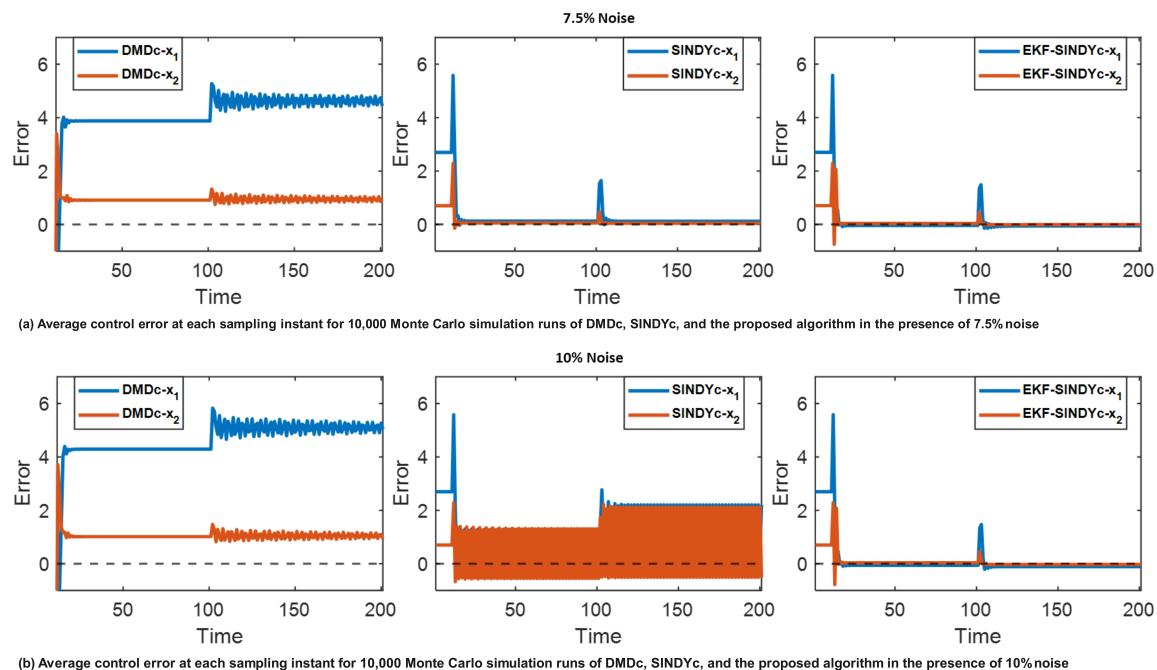
where  $a, b, c, d, e, f, g$  are model parameters. The simulation duration,  $T$ , is defined as 100, with a measurement sampling interval of 0.1. The model sampling interval takes every 5 measurement sampling intervals, which yields  $\Delta t = 0.5$ , and a total of 200 model sampling intervals. For the initial 100 model sampling intervals, the parameter values are set as 3.8, 3, 1, 1.1, 0.5, 0.85, 1 and are adjusted to 4, 2.9, 1, 1, 0.5, 1, 1.2 in the final 100 model sampling intervals. The control objective is to guide the system states towards their respective target values,  $\frac{a}{f_d}$  and  $bg$ .

Owing to the limited sample numbers, in this example, the performance of the DMDc and SINDYc is compared with the proposed algorithm. For the initial system identification, 80 samples are provided at two distinct noise levels, corresponding to 7.5%, and 10% relative to the standard deviation of the clean data. The feature library for SINDYc and the proposed algorithm is

$$\Theta(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{1} & \mathbf{x} & \mathbf{u} & \mathbf{x} \otimes \mathbf{x} & \sin(\mathbf{x}) & \cos(\mathbf{x}) & \tanh(\mathbf{x}) \end{bmatrix}. \quad (32)$$

To rigorously evaluate the algorithms' performance, 10,000 Monte Carlo simulation runs are conducted with  $\mathbf{x}_0 = \begin{bmatrix} 3 & 2 \end{bmatrix}$ ,  $\lambda = 0.4$ , and  $N_c = N_p = 5$ . The MPC weight matrices are set as  $Q = Q_{N_p} = \text{diag}\{0.9 \ 0.9\}$ ,  $R_u = 0.95$ .

Figure 6 shows the average error between the reference trajectories and the state trajectories controlled by the evaluated algorithms. This comparison is based on the 10,000 Monte Carlo simulation runs conducted under two distinct noise levels. It can be observed from Figure 6 (a) and (b) that the DMDc shows larger errors in controlling the states to their corresponding objectives and becomes unstable after the system dynamic changes. The inferior performance of DMDc is attributable to its limited capacity for managing nonlinearities and system dynamic changes. Additionally, while SINDYc can achieve relatively lower control errors, its performance becomes unstable when the noise level increases to 10%, and it exhibits larger control errors after changes in system dynamics.



**Figure 6.** Monte Carlo average error plots based on the control performance of DMDc, SINDYc, and the proposed algorithm.

In contrast, the proposed EKF-based recursive sparse nonlinear digital-twin identification with MPC algorithm consistently demonstrates stable control performance and reduced average error among the three algorithms considered. This result further shows that the proposed algorithm is robust to both system dynamic changes and the initial model defects arising from the training data noise. By including real-time measurements to recursively adjust the digital-twin model online, the proposed algorithm achieves enhanced control performance.

## 6. Conclusions

The digitalization of industrial production processes has emerged as an inevitable trend in recent years. In this procedure, the digital twin plays a significant role, serving as the bridge between the physical system and the digital world. This study focuses on the modeling, synchronization, and interaction perspectives of the digital twin's underlying algorithm. The proposed algorithm first utilizes SINDYc to streamline the initial digital twin model identification procedure, saving the model development time while maintaining model precision. Subsequently, the EKF algorithm is integrated with the SINDYc to recursively update the selection of model features and their corresponding parameters to ensure precise synchronization between the digital twin model and the physical operations. Finally, by integrating the proposed adaptive digital twin identification with the MPC, enhanced control performance is achieved, promoting the effective interaction between user instructions and system dynamics. The comparative results from the industrial case study and simulation examples demonstrate that the proposed algorithm is robust to both training data noise and system dynamic changes, making it well-suited for performing online digital twin identification with control. To realize the complete life cycle of digital twins, further research is necessary to explore the deployment and maintenance of the proposed algorithm in real industrial applications. Integrating the proposed algorithm with the widely applicable advanced automation software and platforms, such as the Microsoft Azure cloud platform, can facilitate the development and implementation of the complete digital twin life cycle in general industrial production environments and will be the future direction of this research.

**Author Contributions:** Conceptualization, J.W. and L.C.; methodology, J.W.; software, J.W.; validation, J.W., L.C.; formal analysis, J.W.; investigation, J.W.; resources, Y.C, R.B.G.; data curation, R.B.G.; writing—original draft

preparation, J.W.; writing—review and editing, J.W., L.C., Y.C, R.B.G; visualization, J.W.; supervision, Y.C, R.B.G; project administration, R.B.G; funding acquisition, R.B.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, under Grant GR001536.

**Conflicts of 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.

## References

- Juarez, M.G.; Botti, V.J.; Giret, A.S. Digital Twins: Review and Challenges. *J. Comput. Inf. Sci. Eng.* **2021**, *21*, 030802.
- Yang, D.; Karimi, H.R.; Kaynak, O.; Yin, S. Developments of digital twin technologies in industrial, smart city and healthcare sectors: a survey. *Complex Eng. Syst.* **2021**.
- Datta, S.P.A. Emergence of Digital Twins. *J. Innovation Manage.* **2017**, *5*, 14–34.
- El Saddik, A. Digital Twins: The Convergence of Multimedia Technologies. *IEEE MultiMedia* **2018**, *25*, 87–92.
- Wang, S.; Jiao, Y.; Wang, L.; Wang, W.; Ma, X.; Xu, Q.; Lu, Z. Research on the Digital Twin System of Welding Robots Driven by Data. *Sensors* **2025**, *25*, 3889.
- Rodrigo, M.S.; Rivera, D.; Moreno, J.I.; Álvarez-Campana, M.; López, D.R. Digital Twins for 5G Networks: A Modeling and Deployment Methodology. *IEEE Access* **2023**, *11*, 38112–38126.
- Grieves, M.W., Digital Twins: Past, Present, and Future. In *The Digital Twin*; Springer International Publishing, 2023; p. 97–121.
- Huang, J.; Huang, S.; Moghaddam, S.K.; Lu, Y.; Wang, G.; Yan, Y.; Shi, X. Deep Reinforcement Learning-Based Dynamic Reconfiguration Planning for Digital Twin-Driven Smart Manufacturing Systems With Reconfigurable Machine Tools. *IEEE Trans. Industr. Inform.* **2024**, *20*, 13135–13146.
- Chancharoen, R.; Chaiprabha, K.; Wuttisittikuljij, L.; Asdornwised, W.; Saadi, M.; Phanomchoeng, G. Digital Twin for a Collaborative Painting Robot. *Sensors* **2022**, *23*, 17.
- Arrano-Vargas, F.; Konstantinou, G. Modular Design and Real-Time Simulators Toward Power System Digital Twins Implementation. *IEEE Trans. Industr. Inform.* **2023**, *19*, 52–61.
- Tao, F.; Xiao, B.; Qi, Q.; Cheng, J.; Ji, P. Digital twin modeling. *Journal of Manufacturing Systems* **2022**, *64*, 372–389.
- Jiang, Y.; Yin, S.; Li, K.; Luo, H.; Kaynak, O. Industrial applications of digital twins. *Philos. Trans. R. Soc. A* **2021**, *379*, 20200360.
- Wang, K.; Tsung, F. Sparse and Structured Function-on-Function Quality Predictive Modeling by Hierarchical Variable Selection and Multitask Learning. *IEEE Trans. Industr. Inform.* **2021**, *17*, 6720–6730.
- Cui, L.; Wang, X.; Liu, D.; Wang, H. An Adaptive Sparse Graph Learning Method Based on Digital Twin Dictionary for Remaining Useful Life Prediction of Rolling Element Bearings. *IEEE Trans. Industr. Inform.* **2024**, *20*, 10892–10900.
- Brunton, S.L.; Proctor, J.L.; Kutz, J.N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *PNAS* **2016**, *113*, 3932–3937.
- Wang, J.; Moreira, J.; Cao, Y.; Gopaluni, B. Time-Variant Digital Twin Modeling through the Kalman-Generalized Sparse Identification of Nonlinear Dynamics. In Proceedings of the 2022 American Control Conf. (ACC), 2022, pp. 5217–5222.
- Jiang, Y.; Yin, S.; Li, K.; Luo, H.; Kaynak, O. Industrial applications of digital twins. *Philos. Trans. A Math. Phys. Eng. Sci.* **2021**, *379*, 20200360.
- Cvok, I.; Soldo, J.; Deur, J.; Ivanovic, V.; Zhang, Y.; Fujii, Y. Model Predictive Control for Automatic Transmission Upshift Inertia Phase. *IEEE Trans. Control Syst. Technol.* **2023**, *31*, 2335–2349.
- Svensen, J.L.; Sun, C.; Cembrano, G.; Puig, V. Model Predictive Control of Urban Drainage Systems Considering Uncertainty. *IEEE Trans. Control Syst. Technol.* **2023**, *31*, 2968–2975.
- Kaiser, E.; Kutz, J.N.; Brunton, S.L. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. R. Soc. A* **2018**, *474*, 20180335.
- Liu, J.; Liu, X.; Vatn, J.; Yin, S. A generic framework for qualifications of digital twins in maintenance. *J. Autom. Intell.* **2023**, *2*, 196–203.
- Simon, D., Optimal state estimation: Kalman, H Infinity, and Nonlinear Approaches; John Wiley & Sonss, 2006; p. 400–407.

23. Qi, W.; Liu, J.; Chen, X.; Christofides, P.D. Supervisory Predictive Control of Standalone Wind/Solar Energy Generation Systems. *IEEE Trans. Control Syst. Technol.* **2011**, *19*, 199–207.
24. Mennemann, J.F.; Marko, L.; Schmidt, J.; Kemmetmuller, W.; Kugi, A. Nonlinear Model Predictive Control of a Variable-Speed Pumped-Storage Power Plant. *IEEE Trans. Control Syst. Technol.* **2021**, *29*, 645–660.
25. Li, H.; Bai, J.; Zou, H.; Yin, X.; Zhang, R. A two-dimensional model predictive iterative learning control based on the set point learning strategy for batch processes. *J. Process Control* **2024**, *133*, 103133.
26. Garcia, M.R.; Pitta, R.N.; Fischer, G.G.; Neto, E.R. Optimizing Diesel Production Using Advanced Process Control and Dynamic Simulation. *IFAC-PapersOnLine* **2014**, *47*, 358–363.
27. Carelli, A.; da Souza, M. GPC Controller Performance Monitoring and Diagnosis Applied to a Diesel Hydrotreating Reactor. *IFAC-PapersOnLine* **2009**, *42*, 976–981.
28. Bandyopadhyay, R.; Alkilde, O.F.; Menjon, I.; Meyland, L.H.; Sahlertz, I.V. Statistical analysis of variation of economic parameters affecting different configurations of diesel hydrotreating unit. *Energy* **2019**, *183*, 702–715.
29. Lindermeir, A.; Kah, S.; Kavurucu, S.; Mühlner, M. On-board diesel fuel processing for an SOFC-APU—Technical challenges for catalysis and reactor design. *Applied Catalysis B: Environment and Energy* **2007**, *70*, 488–497.
30. Neto, E.T.; Imtiaz, S.A.; Ahmed, S.; Gopaluni, R.B., Hybrid model for a diesel cloud point soft-sensor. In *Modelling of Chemical Process Systems*; Elsevier, 2023; p. 271–314.
31. Kinoshita, S. Introduction to Nonequilibrium Phenomena. In *Pattern Formations and Oscillatory Phenomena*; Elsevier, 2013; pp. 1–59.
32. Kulikov, G.Y.; Kulikova, M.V. Accurate Numerical Implementation of the Continuous-Discrete Extended Kalman Filter. *IEEE Trans. Autom. Control* **2014**, *59*, 273–279.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.